

Разработка программного и методического обеспечения для проведения практических занятий по конструктивной теории машин Тьюринга

И. А. Пушкарев¹, А. В. Огородов²

¹ кандидат физико-математических наук, доцент кафедры фундаментальной информатики и прикладной математики, Вятский государственный университет. Россия, г. Киров. E-mail: god_sha@mail.ru

² магистрант, Вятский государственный университет. Россия, г. Киров. E-mail: tepmi@yandex.ru

Аннотация. В статье рассматриваются некоторые специальные аспекты конструктивной теории машин Тьюринга в рамках преподавания математической логики и теории алгоритмов. В частности, преподавание этой части дисциплины требует использования соответствующего специально разработанного программного обеспечения. В данной работе эта проблема анализируется с позиций накопленного опыта преподавания теории алгоритмов студентам специальности «Прикладная математика и информатика» на основе подхода, основанного на рассмотрении «диаграмм Тьюринга», разработанного немецкими учеными под руководством профессора К. Якобса. На предыдущей стадии преподавания для верификации корректности работы построенных диаграмм также использовалось специально разработанное программное обеспечение. В данной работе предлагается программное обеспечение следующего поколения. Выявляются как недостатки программного обеспечения, использованного ранее, так и пожелания и требования к новому программному продукту, который должен в перспективе занять место старого. Статья содержит идеологическое описание нового программного обеспечения, важные подробности технической реализации и интерфейса и описание новых возможностей разработанного продукта, касающихся, в частности, рассмотрения вопросов моделирования многопроцессорных систем и параллельного программирования.

Ключевые слова: библиотека qt, имитатор, верификация, машина Тьюринга, кроссплатформенность, многопроцессорные системы.

Введение

Основная проблема в преподавании раздела теории алгоритмов, посвященного изучению конструктивной теории машин Тьюринга (или любого другого определения алгоритма), состоит в создании возможностей верификации правильности машин Тьюринга, построенных учащимися. Именно конструктивная разработка машин Тьюринга студентами, по существу, является разработкой ими собственного языка программирования, начиная с небольшого набора простейших команд – языка, предназначенного для работы с *воображаемым* устройством. Программирование на сколько-нибудь приличном уровне немислимо без тестирования: правдоподобно выглядящая машина Тьюринга в действительности может оказаться неочевидно неработоспособной в силу наличия малозаметного конструктивного дефекта. При этом тестирование работы на воображаемом устройстве либо является очень трудоемкой задачей, требующей нереального объема ручной работы, либо требует реализации того самого воображаемого устройства в вид компьютерной модели.

Задача построения таковой модели, подходящей для реализации машин Тьюринга в синтаксисе так называемых диаграмм Тьюринга (см. напр., [2]), была в свое время решена А. Б. Белицем (рис. 1).

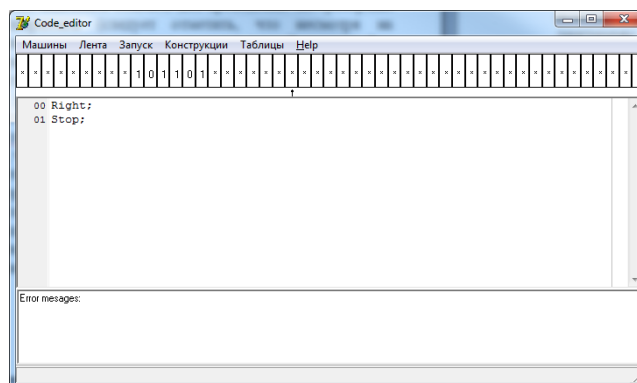


Рис. 1. Основная экранная форма имитатора А. Б. Белица

Имитатор, разработанный А. Б. Белицем, служил надежной основой для проведения лабораторных работ по теории алгоритмов (несмотря на «компьютерное» название, теория алгоритмов является наукой чрезвычайно малопрактичной и неприспособленной для проведения лабораторных работ и даже большого количества практических занятий – за пределами обсуждаемой области).

Следует отметить, что программа, разработанная А. Б. Белицем, не является одной из многочисленных, во множестве выложенных в Интернет примитивных имитаторов, предназначенных для имитации только очень простых машин, записанных в виде «квадратной» таблицы. Она обладает развитым функционалом, позволяющим имитировать работу большого числа постоянно усложняющихся машин, разработанных последовательно, а не как разовые иллюстрационные анимации, притом – разработанных различными пользователями. Однако с ее эксплуатацией, начиная с некоторого момента, возникли определенные проблемы. У имитатора обнаружился целый ряд недостатков, в частности он оказался недостаточно кроссплатформенным.

Тем самым, приобрела известную актуальность задача написания нового имитатора, обладающего свойством кроссплатформенности и, возможно, некоторыми другими полезными для учебного процесса свойствами.

В качестве основы предлагалось переписать имитатор практически с тем же функционалом на основе элегантного способа представления диаграмм Тьюринга (которые сами по себе являются вершинно-помеченными орграфами) в виде некоторого исходного языка программирования, сразу имея в виду по окончании первого этапа работы перейти к расширению и усовершенствованию функционала. Кроме того, предполагалось снабдить разработанную программу методическим обеспечением, облегчающим проведение занятий.

Перечень дополнительных возможностей

В качестве расширения функционала программы предлагаются следующие усовершенствования.

1. Предполагается уменьшить (в идеале – устранить) зависимость работы программы от кодировки символов, замеченную при замене инвентаря дисплейного класса и тем самым существенно повысить кроссплатформенность программы.

2. Существенным недостатком программы А. Б. Белица был ураганный рост длин имен состояний модели машины Тьюринга, получающейся при интерпретации введенной диаграммы. Напомним, что при построении каждой следующей, более сложной, машины Тьюринга, допустимо использование всех уже построенных машин как отдельных операторов, возможно – по несколько экземпляров. При этом состояния построенных машин уже имели довольно длинные названия, к которым теперь приходилось добавлять идентификатор машины и идентификатор номера машины в новой диаграмме. В некоторый момент длины имен переставали помещаться в размер отведенной для них памяти, и корректная компиляция диаграммы становилась невозможна.

Предложить простое автоматическое решение этой проблемы оказалось крайне проблематичным, и был предложен комплекс частичных решений. Именно:

(а) предлагается рассмотреть возможность предоставления пользователю полномочий при необходимости использовать возможность ручного переименования состояний с трансляцией новых имен во все последующие машины;

(б) предложено реализовать два режима имитации машин: условно говоря, «честный» – с буквальным превращением машины Тьюринга в таблицу и пошаговым исполнением с выделением текущей команды (для простых машин) и, напротив «быстрый», когда работа машины имитируется только на экране (для ситуаций, когда на первый план выходит проверка корректности работы).

3. Программа А. Б. Белица работала не с бесконечной в обе стороны, а с закольцованной лентой с переменным количеством ячеек, но (почему-то) с постоянной длиной, поэтому при увеличении длины ленты (которое при необходимости производилось при помощи ручной команды) ячейки становились очень узкими (рис. 2), что было не совсем удобно.

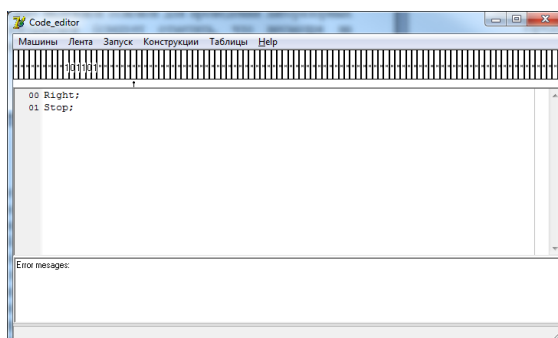


Рис. 2. Эффект сжатия ячеек

4. Разрабатываемая программа является учебно-методической и должна помогать учебному процессу, предлагая иногда (так называемые) «фишки», облегчающие обучение. Например, для понимания того, как работает так называемый «копировщик» (см. далее) достаточно реализовать его для алфавита из двух букв (не считая пробела). Однако в дальнейшем предполагается написание более сложных машин, для которых необходимы копировщики, работающие над более сложными алфавитами. Заставлять учащегося писать отдельно эти копировщики контрпродуктивно, поэтому в программе предусмотрена возможность автоматического создания (с последующим использованием) таких «многобуквенных» копировщиков.

5. Кроме того, программа А. Б. Белица не предоставляла возможностей изучения машин Тьюринга с разными вариантами многопроцессорности (многоленточность, наличие нескольких кареток на одной ленте и другие), не составляющего большой трудности в реализации и очень уместное в силу (случившегося позднее) определения профиля магистратуры как ориентированного на параллельное и многопроцессорное программирование.

Дополнительное методическое обеспечение

1. Естественным образом, в методическое обеспечение проведения занятий входит разработка функционала справочника, встроенного в программу, содержащего исчерпывающее описание синтаксиса, снабженное примерами.

2. В качестве второй дополнительной цели работы естественным образом возникла задача унификации и разумного упорядочения списка задач, предлагаемых учащимся. В ходе проведения занятий учащимся предлагалось большое количество задач, связанных с комбинаторикой слов и вычислениями в унарной, двоичной и троичной позиционных системах счисления.

Задачи эти выдавались учащимся далеко не в случайном порядке, однако поиск оптимального порядка расположения задач как таковой не проводился, и это тоже оказалось задачей, заслуживающей внимания.

Кроме того, естественной задачей представлялась выдача примеров требуемых машин Тьюринга учащимся не сразу, в готовом виде, а после выполнения ими некоторого «норматива», однако от этой идеи «умного» электронного учебника пришлось отказаться в силу трудноформализуемости этого самого «норматива» как условия того, что учащийся самостоятельно попробовал построить машину и она у него «почти заработала» или просто заработала, а «ручное» сообщение преподавателем пароля при демонстрации работающей машины представилось неоправданно сложной и чрезмерно криптографической процедурой.

Основные заметки о программном обеспечении

Основная структура программного обеспечения схожа с предыдущим интерпретатором. Имеется возможность исполнения машины по шагам с подсветкой текущей строки либо же мгновенное исполнение. Также этап запуска разделен на две части: сборка и исполнение.

Ошибка на любом из этапов приводит к невозможности запуска машины. При возникновении какой-либо ошибки информация о ней выводится пользователю, что значительно упрощает процесс создания и отладки.

Для обеспечения переносимости между основными операционными системами (Microsoft Windows и Linux) было принято решение использовать библиотеку Qt, что позволяет, помимо переносимости итоговой программы, разработать полностью адаптивный пользовательский интерфейс. Основной графический интерфейс представлен на рис. 3.

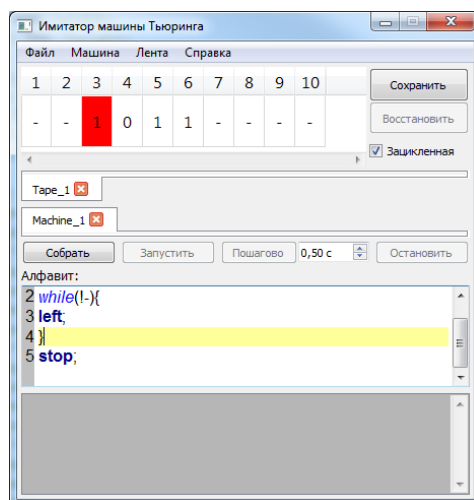


Рис. 3. Адаптивный графический интерфейс

Текущее положение печатающей головки на ленте отображается посредством цветовой подсветки соответствующей ячейки ленты красным цветом. Лента может быть как зацикленной, так и конечной. При достижении границы ленты любая машина будет остановлена с ошибкой, информация о которой будет выведена в нижнюю область интерфейса. Также в качестве дополнительной возможности реализован механизм запоминания текущей ленты и в случае необходимости восстановления ее состояния из ранее запомненного.

Для обеспечения наглядности также введена подсветка ключевых слов и выражений псевдо-языка.

Приложение позволяет выполнять машину как мгновенно, так и с устанавливаемым периодическим шагом по времени. При использовании пошагового выполнения текущая строка подсвечивается в редакторе, что делает процесс более наглядным.

Рассмотрим работу программы на примере машины «Копировщик». Задача данной машины – скопировать последовательность символов правее ее текущего положения.

Диаграмма копировщика представлена на рис. 4.

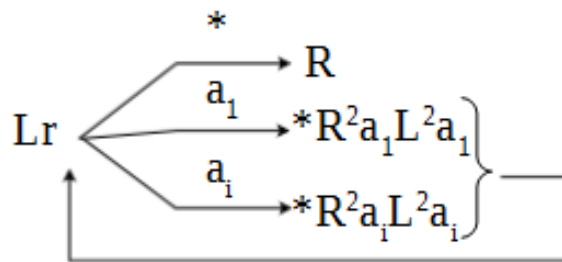


Рис. 4. Диаграмма копировщика

Фрагмент последовательностей команд, требуемых для реализации копировщика на символах '*012', представлен в таблице ниже.

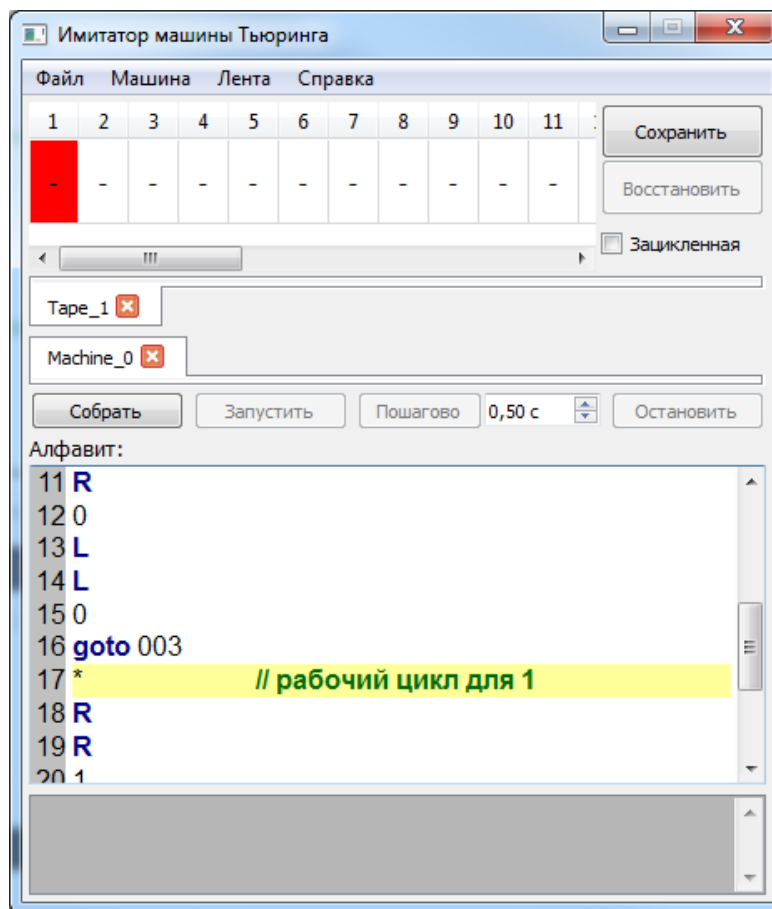


Рис. 5. Реализация машины «копировщик»

000	Stop
001	L
002	r
003	case * goto 004 0 goto 005 1 goto 2 goto
004	R % выход
005	* % рабочий цикл для 0
006	R
007	R
008	0
009	L
010	L
011	0
012	goto 003
013	* % рабочий цикл для 1
014	R
015	R
016	1
017	L
018	L
019	1
020	Goto 003

Результат работы копировщика в двух режимах интерпретации представлен на рис. 6.

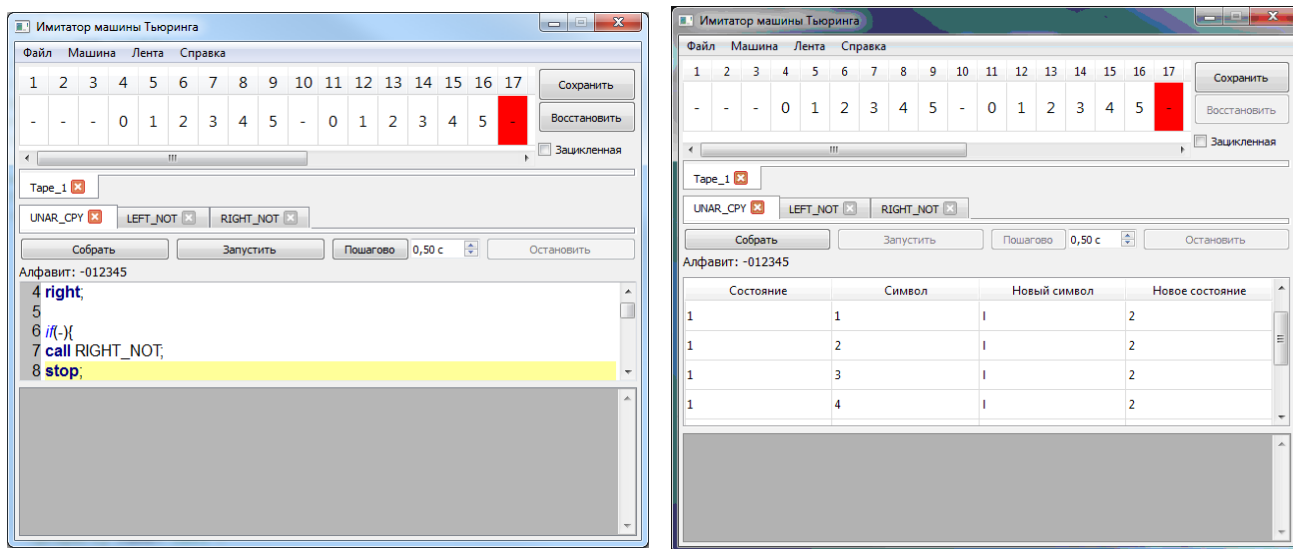


Рис. 6. Результат работы копировщика на алфавите '-012345'

В результате работы встроенного компилятора программа, соответствующая диаграмме (при работе в режиме «честной» имитации), конвертируется в таблицу состояний машины. Соответствующая экранная форма изображена на рис. 7.

Состояние	Символ	Новый символ	Следующее состояние
1	1		2
1	2		2
1	3		2
1	4		2
1	5		2
2	*	*	3
2	0	0	1
2	1	1	1
2	2	2	1

Рис. 7. Таблица состояний машины

Экранная форма автоматического построения и использования копировщика, работающего над другими алфавитами, представлена на рис. 8.

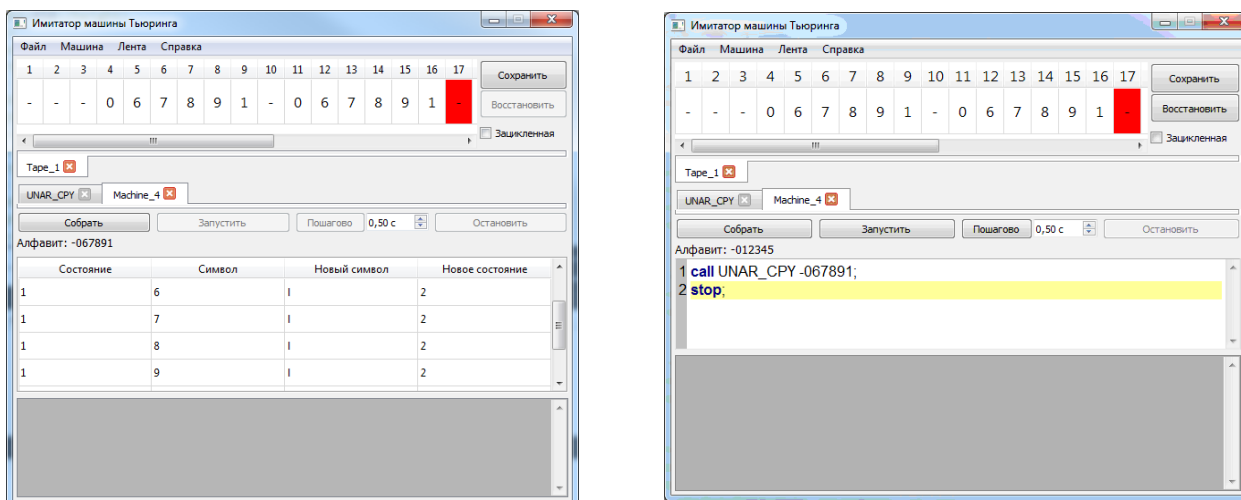


Рис. 8. Результат работы копировщика с переопределенным алфавитом

Многопроцессорные режимы

В заключение проиллюстрируем возможности имитации работы одноленточной многокаре- точной (рис. 9) и многоленточной (рис. 10) машин.

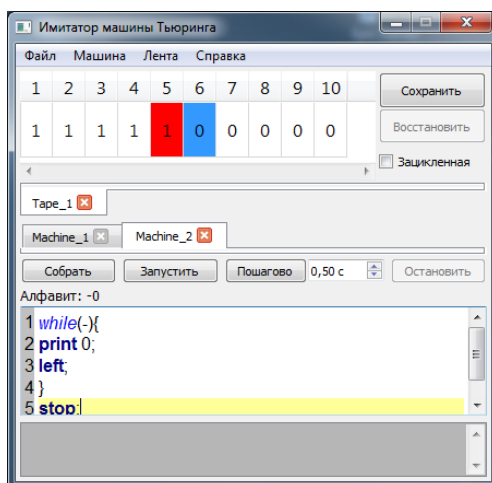


Рис. 9. Результат работы машины с двумя каретками на одной ленте

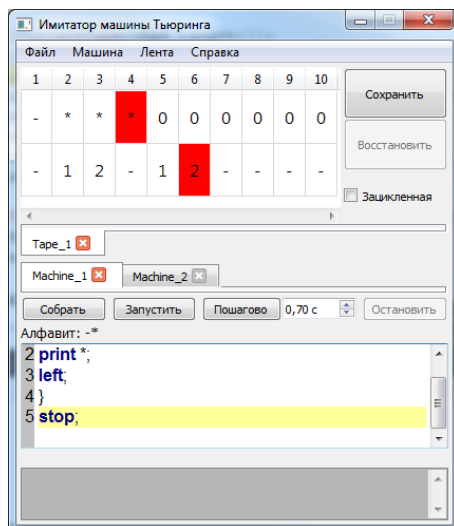


Рис. 10. Машина, работающая на двух различных лентах

Список литературы

1. Павловская Т. А. C/C++. Программирование на языке высокого уровня. СПб. : Питер, 2003. 461 с.
2. Эббингауз Г.-Д. Машины Тьюринга и рекурсивные функции. М. : «Мир», 1972.
3. Qt Documentation. URL: <http://doc.qt.io/> (дата обращения: 14.07.2017).

Development of software and methodological support for practical training on the constructive theory of Turing machines

I. A. Pushkarev¹, A. V. Ogorodov²

¹ PhD of physical and mathematical sciences, associate professor of fundamental Informatics and applied mathematics, Vyatka State University. Russia, Kirov. E-mail: god_sha@mail.ru

² master's degree, Vyatka State University. Russia, Kirov. E-mail: tepmi@yandex.ru

Abstract. The article deals with some special aspects of the constructive theory of Turing machines in the framework of teaching mathematical logic and theory of algorithms. In particular, the teaching of this part of the discipline requires the use of appropriate specially developed software. In this paper, this problem is analyzed from the standpoint of the accumulated experience of teaching the theory of algorithms to students of the specialty "Applied mathematics and Informatics" on the basis of the approach based on the consideration of "Turing diagrams" developed by German scientists under the guidance of Professor K. Jacobs. At the previous stage of teaching, specially developed software was also used to verify the correctness of the diagrams. In this work, a software of the next generation is proposed. The article reveals both the shortcomings of the software used earlier, and the wishes and requirements for a new software product, which should in the future take the place of the old one. The article contains an ideological description of the new software, important details of technical implementation and interface and description of new features of the developed product, in particular, concerning the consideration of issues of modeling of multiprocessor systems and parallel programming.

Keywords: qt library, simulator, verification, Turing machine, cross-platform, multiprocessor systems.

References

1. Pavlovskaya T. A. C/C++. *Programmirovanie na yazyke vysokogo urovnya* [C/C++. Programming in a high-level language]. SPb. Piter. 2003. 461 p.
2. Ebbinghouse G.-D. *Mashiny T'yuringa i rekursivnye funktsii* [Turing Machines and recursive functions]. M. "Mir". 1972.
3. Qt Documentation. Available at: <http://doc.qt.io/> (accessed: 14.07.2017).