

К ДИДАКТИКЕ ТЕОРИИ АЛГОРИТМОВ: МОДИФИКАЦИЯ МАШИНЫ ПОСТА С ЦЕЛЬЮ ОБЕСПЕЧЕНИЯ ВОЗМОЖНОСТИ ИХ СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ

© 2024. М. П. Загорный, И. В. Кандаева
ФГБОУ ВО «Донецкий государственный университет»

В статье рассматривается фрагмент содержания обучения теории алгоритмов - программирование машин Поста. Отмечена принципиальная невозможность программирования машин Поста в структурном стиле. Предложена модификация машин Поста, обеспечивающая возможность их структурного программирования. Сделан вывод о целесообразности введения в содержание обучения теории алгоритмов ознакомления обучающихся с модифицированными машинами Поста.

Ключевые слова: обучение теории алгоритмов, машины Поста, структурное программирование, модификация машин Поста.

Введение. В настоящее время профессии, требующие знания математической логики и теории алгоритмов, являются весьма востребованными [1]. Обучение теории алгоритмов нельзя считать предметом одной лишь дидактики математики. Обучение теории алгоритмов, по нашему мнению, должно рассматриваться как междисциплинарная практика и как междисциплинарная область научных исследований. Сложность таких исследований и практик отмечена, например, в работе [2].

Одним из нежелательных следствий указанной сложности является отрицательный дидактический эффект частичного сокрытия динамики развития математического знания от обучающихся [3]. Это чревато опасностью одностороннего, формального усвоения обучающимися учебного материала, возможностью возникновения в сознании обучающихся стойкого убеждения в том, что исследования в фундаментальных областях математики и информатики давно завершены, сами основания указанных наук в ревизии не нуждаются (и нуждаться в принципе не могут). Такое убеждение является не только наивным, но и вполне может вызывать тревогу – его наличие может быть заметным препятствием на пути формирования творческого отношения обучающихся к содержанию обучения. В работах [4; 5] мы уже обращались к этой проблеме в контексте содержания обучения нейроинформатике и математической логике.

Важным элементом содержания обучения теории алгоритмов является освоение способов формализации алгоритмов. Один из таких способов – машины Поста [6]. Когда студенты осваивают этот способ, наряду с его несомненными достоинствами становится очевидным и его существенный дидактический недостаток. Он заключается в том, что естественный для машин Поста метод реализации парадигмы структурного программирования – получение новой машины путем композиции ранее сконструированных машин – предполагает внесение изменений в коды программ тех машин, которые подвергаются композиции. Сама конструкция классических машин Поста не допускает повторного использования ранее созданного программного кода в новом контексте без внесения изменений. Формализация сколько-нибудь сложных алгоритмов в виде классических машин Поста становится практически неосуществимой по причине неудобочитаемости программного кода, неизбежно

приобретающего весьма запутанный «спагетти-вид». Проблема повышения удобочитаемости кода программ для абстрактных вычислительных машин (в контексте машин Тьюринга) путем использования в синтаксисе команд «говорящей» расширенной нотации как эвристического приема уже обсуждалась нами в [7].

Целью данной работы является представление предлагаемой нами небольшой модификации классических машин Поста (именуемых далее машинами МП-1), введение которой приводит к появлению возможности составлять программы для модифицированных машин Поста (именуемых далее машинами МП-2) в виде главных модулей, обращающихся к вызываемым ими подпрограммам. При этом код каждой подпрограммы, будучи единожды разработан, не нуждается для повторного использования во внесении каких-либо изменений.

Таким образом, у обучающегося появляется возможность программировать машины Поста структурно, формализовать в понятном виде относительно более глубокие алгоритмические идеи, что, по нашему мнению, должно способствовать повышению уровня мотивации учения при освоении обсуждаемого аспекта теории алгоритмов. Отметим также, что каждая программа для машины МП-1 является также и корректной программой для машины МП-2: принцип совместимости «снизу вверх» предлагаемой нами модификацией не нарушается.

Основная часть. Машина МП-2 (как и машина МП-1) имеет ленту, над которой движется каретка. На ленту помещаются входные данные, на ней же образуются промежуточные и выходные данные. Каретка – это автомат, преобразующий данные. Машина работает под управлением программы, составляемой человеком. Две различные машины Поста МП-2 (как и две различные машины МП-1) отличаются только программами. Сконструировать машину Поста для решения некоторой массовой задачи – значит составить для нее программу.

Лента машины МП-2 (как и машины МП-1) – это конечная последовательность ячеек. Каждая из ячеек может быть помечена или не помечена. Если условиться помеченную ячейку обозначать символом 1, а непомеченную – символом 0, то любое состояние ленты можно представить словом в алфавите $\{0, 1\}$. Например, слово 001110 представляет ленту, состоящую из шести ячеек, содержащую сплошной непустой массив помеченных ячеек длины 3.

Каретка на каждом такте работы машины МП-2 (и машины МП-1) находится над одной и только над одной ячейкой ленты. Об этой ячейке говорят, что она обозревается кареткой. Если условиться обозначать каретку символом q и записывать его перед символом, изображающим обозреваемую ячейку, то любое состояние ленты вместе с указанием положения каретки может быть представлено словом в алфавите $\{0, 1, q\}$. Например, слово $00q1110$ соответствует представленной выше ленте, третья слева ячейка которой обозревается кареткой.

Машина МП-2 имеет регистр IP (Instruction Pointer). Можно считать, что машина МП-1 тоже имеет такой регистр. У машины МП-1 регистр IP является одноместным и либо пуст, либо содержит некоторое натуральное число – номер очередной исполняемой команды. Перед первым тактом работы машины МП-1 регистр IP всегда содержит номер 1. Регистр IP машины МП-2 – не одноместный, а стековый. Стек либо пуст, либо содержит одно или несколько слов. Слово, находящееся на вершине стека IP машины МП-2 – это имя очередной исполняемой команды. Перед первым тактом работы машины МП-2 стек IP всегда содержит единственное слово 1. Если условиться содержимое регистра IP (как машины МП-1, так и машины МП-2) записывать в виде нижнего правого индекса при символе q , разделяя при необходимости компоненты

содержимого пробелами, то полная информация о любом состоянии машины МП-2 (как и машины МП-1) в целом может быть представлена некоторым словом (слова этого рода используются также для полного представления состояний машин Тьюринга и называются словами Поста). Например, слово Поста $00q_51110$ сообщает о том, что на некотором такте работы машины каретка обозревает третью слева ячейку шестиячеечной ленты. Если речь идет о машине МП-1, то указанное слово сообщает также, что регистр IP не пуст, но содержит натуральное число (номер) 5. Если же относить это слово к машине МП-2, то оно сообщает, что стек IP не пуст, содержит единственное значение, этим значением является слово 5.

Можно говорить, что словами Поста однозначно описываются конфигурации машин Поста (как МП-1, так и МП-2). Процесс работы машины (и МП-1, и МП-2) можно представлять как последовательность сменяющих одна другую конфигураций. Смена конфигураций происходит в результате выполнения составляющих программу команд. Для машины МП-2 имеется семь типов команд. Для машин МП-1 число типов равняется шести.

Команды сдвига влево у машин МП-2 имеют имя (у машин МП-1 – номер), оператор L и операнд-слово (у машин МП-1 – операнд-номер). Если значение на вершине стека IP (у машин МП-1 – значение в регистре IP) совпадает с именем (у машин МП-1 – с номером) команды – она выполняется: каретка смещается на одну позицию влево, значение на вершине стека IP (у машин МП-1 – значение в регистре IP) заменяется значением операнда. Если до начала выполнения команды каретка обозревала самую левую ячейку ленты, то к ленте слева добавляется непомеченная ячейка, которая становится обозреваемой кареткой. Например, командой $5. L 24$; конфигурация $00q_51110$ переводится в конфигурацию $0q_{24}01110$. Команды сдвига вправо имеют аналогичный смысл (с точностью до замены левого на правое). Их оператор – R .

Команды установки метки у машин МП-2 имеют имя (у МП-1 – номер), оператор V и операнд-слово (у машин МП-1 – операнд-номер). Результатом выполнения команды является появление метки в ранее непомеченной ячейке, обозреваемой кареткой, и замена значения на вершине стека IP (у машин МП-1 – значения в регистре IP) значением операнда. Если ячейка, обозреваемая кареткой, уже была помеченной, то происходит аварийный останов машины. Например, командой $2. V 7$; конфигурация $0q_201$ переводится в конфигурацию $0q_711$. Команды удаления метки имеют аналогичный смысл (но происходит удаление метки). Их оператор – X . При попытке удалить метку в уже непомеченной ячейке происходит аварийный останов.

Команды условной передачи управления имеют имя (у МП-1 – номер), оператор $?$ и два операнда-слова (у машин МП-1 – два операнда-номера). Результат выполнения команды: если обозреваемая кареткой ячейка не помечена, то значение на вершине стека IP (у машин МП-1 – значение в регистре IP) заменяется значением первого операнда, в противном случае – второго операнда. Например, командой $3. ? 1 8$; конфигурация $0q_300$ переводится в конфигурацию $0q_100$, а конфигурация $0q_310$ – в конфигурацию $0q_810$.

Команды опустошения стека (у машин МП-1 они называются стоп-командами) имеют имя (у машин МП-1 – номер), оператор S и не имеют операндов. Результатом выполнения команды является снятие с вершины стека IP (у машин МП-1 – удаление из регистра IP) одного значения. Когда IP пуст, происходит нормальный останов машины (и МП-1, и МП-2). Данные, имеющиеся на ленте после нормального останова машины, рассматриваются как выходные данные.

Команды помещения значения в стек IP (им нет аналога у машин МП-1) имеют имя, оператор P и два операнда-слова. Результат выполнения: с вершины стека IP снимается одно слово, в стек вталкивается значение первого операнда, над ним на вершину стека помещается значение второго операнда. Например, командой $1. P 2 1-S$ конфигурация $00q_11110$ переводится в $00q_{2 1-S}1110$. Смысл введения команд этого типа таков: управление выполнением программы передается подпрограмме; для возврата управления из подпрограммы в вызвавший ее модуль достаточно срабатывания в подпрограмме команды опустошения стека. При этом подпрограммы могут быть разработаны независимо от вызывающих их модулей, одна и та же подпрограмма может вызываться различными модулями из различных точек, коды подпрограмм (что очень важно), будучи единожды составлены, не нуждаются для использования подпрограмм во внесении каких-либо изменений. Говоря кратко, именно путем введения для машин МП-2 команд помещения значения в стек (наряду с небольшой модификацией стоп-команд машины МП-1 и превращения их в команды опустошения стека) достигается возможность при программировании машин МП-2 реализации идей структурного программирования. При этом (синтаксически, знаково-символьно) каждая корректно записанная программа для машины МП-1 оказывается корректно записанной программой для машины МП-2, чем обеспечивается совместимость машин МП-1 и МП-2 «снизу вверх».

Одной из причин высокой трудоемкости разработки программ для машин Поста (что может приводить к некоторому снижению уровня мотивации учения обучающихся) является тот факт, что машина Поста – машина воображаемая. Человеку приходится и составлять программы, и имитировать их выполнение, воображая машины и фактически «работая за них». Для устранения указанной трудности нами разработан имитатор «Машина Поста МП-2» [8]. Его использование предполагает, что человек составляет программу, формирует входные данные и запускает процесс выполнения программы машиной. У человека имеется возможность наблюдать выполнение программы в пошаговом режиме (то есть видеть все конфигурации машины – от начальной до конечной) или позволять машине, не останавливаясь, выполнить программу «до конца» (рис. 1).

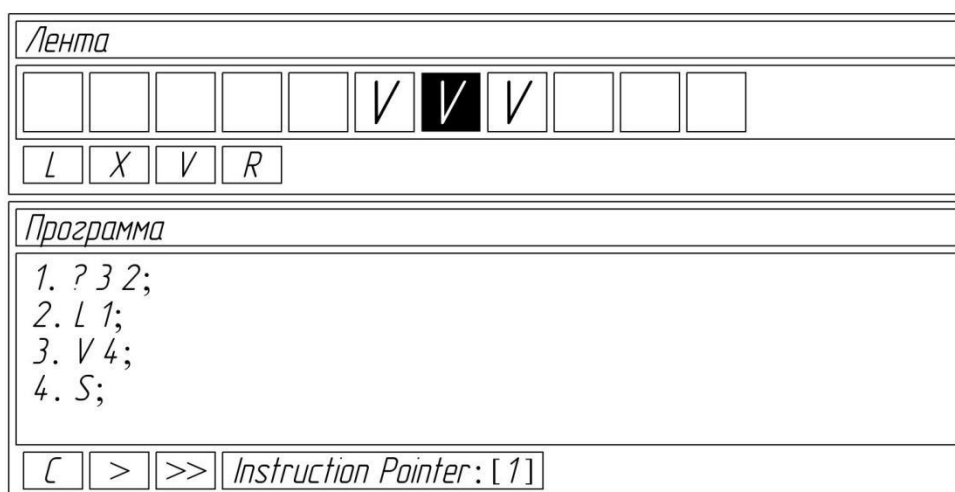


Рис. 1. Схематическое изображение рабочего окна имитатора «Машина Поста МП-2»

С помощью кнопок *L*, *X*, *V* и *R* человек формирует входные данные. Нажатие кнопки *L* смещает каретку на одну позицию влево (кнопки *R* – на одну позицию вправо). Нажатие кнопки *V* приводит к появлению метки в непомеченной ячейке, обозреваемой кареткой. Если каретка обозревает помеченную ячейку, то нажатием кнопки *X* метка, содержащаяся в ячейке, удаляется – ячейка становится непомеченной. Текстовое поле «Программа» служит для введения текста (кода) программы. Нажатием кнопки *S* машина приводится в состояние готовности к пуску – стек *IP* становится содержащим единственное значение 1. Нажимая кнопку *>*, можно наблюдать за тем, как машина в пошаговом режиме выполняет команды в соответствии с программой и как при этом конфигурации машины сменяют одна другую. Если нажать кнопку *>>*, то машина будет работать без остановок – пока не случится аварийный или нормальный останов. Завершение работы аварийным остановом или отсутствие завершения (вход в бесконечный цикл) является признаком некорректности программы. Корректная программа должна завершать работу машины нормальным остановом.

Приведем примеры (довольно искусственные, не претендующие на оптимальность решения, поскольку мы хотим продемонстрировать саму возможность) структурного программирования машины МП-2 (что невозможно для машин МП-1).

Пример 1: пусть известно, что перед началом работы машины на ленте имеется сплошной непустой массив помеченных ячеек. Вне пределов массива, быть может, имеются непомеченные ячейки. Каретка вначале обозревает одну из ячеек массива. После нормального останова длина массива должна быть больше исходной на единицу. Составленная в структурном стиле программа может быть такой:

1. P 2 1-A; 1-A. L 2-A;
2. S; 2-A. ? 3-A 1-A;
- 3-A. V 4-A;
- 4-A. S;

Главный модуль (команды первого столбца) первой командой вызывает подпрограмму *A*. Команды подпрограммы *A* (второй столбец) приводят каретку к левому краю массива помеченных ячеек и побуждают пометить пустую ячейку, непосредственно слева примыкающую к массиву. Затем управление возвращается главному модулю. Получив управление, главный модуль второй командой полностью опустошает стек *IP*, что приводит к нормальному останову. Все действия, соответствующие условию задачи, выполняются подпрограммой *A*. Роль главного модуля сводится к вызову подпрограммы *A* и обеспечению нормального останова машины после возврата управления подпрограммой.

Пример 2: пусть известно, что перед началом работы машины на ленте имеется сплошной непустой массив помеченных ячеек. Слева от массива имеется не менее одной непомеченной ячейки. Справа от массива, быть может, также имеются непомеченные ячейки (хотя их может и не быть). Каретка вначале обозревает одну из непомеченных ячеек, находящихся левее массива. После нормального останова длина массива должна быть больше исходной на единицу. Составленная в структурном стиле программа может быть такой:

1. P 2 1-B; 1-B. R 2-B;
2. S; 2-B. ? 1-B 3-B;
- 3-B. L 4-B;
- 4-B. V 5-B;
- 5-B. S;

Главный модуль (первый столбец) первой командой вызывает подпрограмму В. Команды подпрограммы В (второй столбец) ведут каретку вправо к левому краю массива помеченных ячеек. Когда каретка достигает первой помеченной ячейки, она смещается влево и помечает пустую ячейку, после чего управление возвращается подпрограммой главному модулю. Роль главного модуля, как и в первом примере, сводится лишь к вызову «работающей» подпрограммы и обеспечению нормального останова после возврата управления.

Пример 3: пусть известно, что перед началом работы машины на ленте имеется сплошной непустой массив помеченных ячеек. Массив либо занимает всю ленту, либо вне его пределов имеются непомеченные ячейки. Каретка вначале обзрывает ячейку, находящуюся не правее массива. После нормального останова длина массива должна быть больше исходной на единицу.

Ясно, что решение этой задачи сводится к использованию результатов решения двух предыдущих задач. Главный модуль должен проверить, помечена ли первоначально обзриваемая кареткой ячейка. Если помечена – вызвать подпрограмму А, в противном случае – подпрограмму В. После возврата управления той подпрограммой, которая будет вызвана, достаточно обеспечить нормальный останов. Составленная в структурном стиле программа может быть такой:

1. ? 2 3;	1-A. L 2-A;	1-B. R 2-B;
2. P 4 1-B;	2-A. ? 3-A 1-A;	2-B. ? 1-B 3-B;
3. P 4 1-A;	3-A. V 4-A;	3-B. L 4-B;
4. S;	4-A. S;	4-B. V 5-B;
		5-B. S;

Нетрудно заметить, что при решении последней задачи вновь разработан был лишь главный модуль, а коды подпрограмм А и В были использованы повторно без какого-либо изменения.

Заключение. Мы считаем, что после традиционного освоения изучающими теорию алгоритмов классических машин Поста (МП-1) целесообразным является (быть может, факультативное) ознакомление обучающихся и с модифицированными предложенным нами способом машинами Поста (машинами МП-2). Это позволяет формировать и развивать у обучающихся стремление к программированию в структурном стиле, мотивировать учение, вызывая интерес к выражению элементарными средствами нетривиальных алгоритмических мыслей.

СПИСОК ЛИТЕРАТУРЫ

1. Каменецкий С. Ю. Учет дидактических концепций при использовании многоуровневых мобильных компьютерных задачников в обучении разделу «Логика» школьного курса информатики / С. Ю. Каменецкий // Вестник РУДН. Серия: Информатизация образования. – 2019. – Т. 16. – № 1. – С. 64-72.
2. Осмоловская И. М. О методологии междисциплинарных исследований в дидактике / И. М. Осмоловская // Ценности и смыслы. – 2022. – № 6(82). – С. 20-29.
3. Когаловский С. Р. Читая Ильенкова (Как школа должна учить мыслить?) / С. Р. Когаловский // Вестник Ивановского государственного университета. Серия «Гуманитарные науки». – 2021. – Вып. 1. – С. 122-133.
4. Загорный М. П. Дополнение к содержанию обучения нейроинформатике: нейросети Калбертсона / М. П. Загорный // Донецкие чтения 2023: образование, наука, инновации, культура и вызовы современности: Материалы VIII Международной научной конференции (Донецк, 25-27 октября 2023 г.). – Том 6: Педагогические науки. Часть 3 / под общей редакцией проф. С. В. Беспаловой. – Донецк: Изд-во ДонГУ, 2023. – С. 258-261.

5. Зарипов Е. В. К дидактике математической логики: опыт разработки студентом оригинального производного правила вывода формализованного исчисления высказываний / Е. В. Зарипов, М. П. Загорный // Современный учитель: профессиональная компетентность и социальная значимость: Материалы III Международной научно-практической конференции (Донецк, 27 июня 2024 г.). – Том 3 / под общей редакцией И. А. Кудрейко. – Донецк: Изд-во ДонГУ, 2024. – С. 42-46.
6. Успенский В. А. Машина Поста / В. А. Успенский. – М.: Наука, 1979. – 96 с.
7. Загорный М. П. К дидактике теории алгоритмов: расширенная нотация внутренних состояний машины Тьюринга как эвристический прием / М. П. Загорный // Донецкие чтения 2019: образование, наука, инновации, культура и вызовы современности: Материалы IV Международной научной конференции (Донецк, 31 октября 2019 г.). – Том 6: Педагогические науки. Часть 1 / под общей редакцией проф. С. В. Беспаловой. – Донецк: Изд-во ДонНУ, 2019. – С. 96-99.
8. Машина Поста МП-2 [Электронный ресурс] – URL: <http://docsagmax.ru/post-machine/pm-2.html>.

Поступила в редакцию 16.11.2024 г.

TO THE DIDACTICS OF THE ALGORITHMS THEORY: MODIFICATION OF POST'S MACHINES IN ORDER TO ENSURE THEIR STRUCTURAL PROGRAMMING OPPORTUNITIES

M. P. Sagorny, I. V. Kandaeva

The passage of the content of the algorithms theory teaching – Post's machines programming – is considered. The principal impossibility of Post's machines programming in the structural style is noted. The modification of Post's machines giving the opportunity of their structural programming is suggested. The conclusion about the feasibility of introducing students familiarization with Post's modified machines in the algorithms theory content teaching is made.

Key words: algorithms theory teaching, Post's machines, structural programming, modification of Post's machines

Загорный Максим Петрович

Начальник Управления дополнительного образования, ФГБОУ ВО «Донецкий государственный университет», г. Донецк, ДНР, РФ.
E-mail: udo@donnu.ru

Sagorny Maxim Petrovich

Head of the Department of Additional Education, FSBEI HE «Donetsk State University», Donetsk, DPR, RF.
E-mail: udo@donnu.ru

Кандаева Ирина Васильевна

Старший преподаватель, ФГБОУ ВО «Донецкий государственный университет», г. Донецк, ДНР, РФ.
E-mail: i.kandaeva@donnu.ru

Kandaeva Irina Vasilievna

Senior lecturer, FSBEI HE «Donetsk State University», Donetsk, DPR, RF.
E-mail: i.kandaeva@donnu.ru