

Виртуальные машины

Введение

Свыше сорока лет назад выдающийся американский математик Эмиль Л. Пост опубликовал в «Журнале символической логики» статью «Финитные комбинаторные процессы, формулировка !» (ее перевод составляет приложение к основному тексту настоящей брошюры). В этой статье и в появившейся одновременно в «Трудах Лондонского математического общества» статье английского математика А. М. Тьюринга «О вычислимых числах с приложением к проблеме разрешения» были даны первые уточнения понятия «алгоритм» — одного из центральных понятий математической логики и кибернетики,— начинающего играть все более и более важную роль в вопросах автоматизации, а поэтому и во всей жизни современного общества.

Уточнения понятия «алгоритм», предложенные Постом и Тьюрингом, не теряют своего значения до нашего времени. Машина Тьюринга постоянно используется в качестве рабочего аппарата в современной теории алгоритмов; машина Поста менее популярна, хотя — или именно по той причине, что — она проще машины Тьюринга). Указанные работы замечательны еще и тем, что в них за несколько лет до появления первых действующих экземпляров больших (так называемых «универсальных») вычислительных машин (сперва даже не электронных, а электромеханических) были в абстрактной форме предвосхищены основные принципиальные черты таких машин. Сами конструкции, предложенные Постом и Тьюрингом, сформулированы ими в виде неких «абстрактных машин»; это сделано в явной форме Тьюрингом и в неявной — Постом, у которого термин «машина» отсутствует. Изложение построений Тьюринга часто приводится в литературе по теории алгоритмов, в том числе и популярной. Что же касается построений Поста, то, несмотря на их даже большую, чем тьюринговские, простоту, они, за исключением оригинальной статьи Поста, долгое время не излагались ни в специальной, ни в популярной литературе). Во то же время именно эти построения, как показывает опыт осуществлявшегося автором преподавания, могут составить не менее, чем машины Тьюринга, естественное введение в теорию алгоритмов.

Машина Поста

«Внешний вид» машины Поста

Прежде всего предупредим читателя, что машина Поста не есть реально существующее, сделанное кем-то устройство; поэтому слова «внешний вид» и взяты в кавычки. Машина Поста, как и ее близкий родственник—машина Тьюринга, — представляет собой мысленную конструкцию, существующую лишь в нашем воображении) (хотя ее в принципе и можно было бы изготовить «в металле»)). Именно это имеют в виду, когда говорят о машинах Поста и Тьюринга, что они суть «абстрактные» вычислительные машины. Однако для нас будет несущественным, *что* машины Поста на самом деле нет. "Напротив, мы будем предполагать ее «как бы существующей» — для наглядности. И подобно тому, как можно выучиться считать на счетах или на логарифмической линейке, не имея перед собой этих приборов, а пользуясь лишь их описаниями и представляя их себе мысленно, так же и мы научимся вычислять на машине Поста, прилагая наше воображение к тому ее описанию, которое сейчас будет дано.

Машина Поста состоит из *ленты* и *каретки* (называемой также *считывающей и записывающей головкой*). Лента бесконечна и разделена на *секции* одинакового размера: для наглядности ленту будем считать расположенной горизонтально (рис. 1).

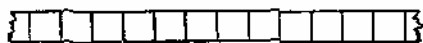


Рис. 1. Лента машины Поста разделена на секции и неограниченно простирается влево и вправо.

Бесконечность ленты находится в противоречии со сделанным выше утверждением, что машину Поста можно было бы в принципе построить. Дело в том, что мы объявили ленту *бесконечной* лишь для простоты изложения. С тем же успехом можно было бы предположить, что лента не бесконечная, а лишь *неограниченно растущая* в обе стороны: например, можно было бы считать, что лента наращивается на одну секцию, как только каретка доходит до конца ленты и должна двигаться дальше (о движении каретки смотря ниже), или считать, что за каждую единицу времени слева и справа нарастает по одной секции. Нам,

однако, будет удобнее считать, что все секции слева и справа уже выросли, и тем самым, хотя и в ущерб реальности, полагать ленту бесконечной в обе стороны.

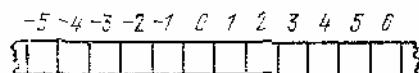


Рис. 2.

Порядок, в котором расположены секции ленты, подобен порядку, в котором расположены все целые числа. Поэтому естественно ввести на ленте «целочисленную систему координат», занумеровав секции целыми числами $\dots, -3, -2, -1, 0, 1, 2, 3, \dots$ (рис. 2).

Мы будем считать, что система координат жестко сопоставлена с лентой, и получим таким образом возможность указывать какую-либо секцию ленты, называя ее порядковый номер или координату. (Иногда, впрочем, бывает удобно наряду с основной, «постоянной» системой координат, ввести еще вспомогательную, «временную», сдвинутую по отношению к первоначальной.)

В каждой секции ленты может быть либо ничего не записано (такая секция называется *пустой*), либо записана *метка* u (тогда секция называется *отмеченной*) (рис. 3),

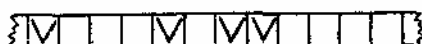


Рис. 3. В каждой секции ленты либо не записано ничего, либо записана метка.

Информация о том, какие секции пусты, а какие отмечены, образует *состояние ленты*. Иными словами, состояние ленты — это распределение меток по ее секциям). Как мы далее увидим, состояние ленты меняется в процессе работы машины.

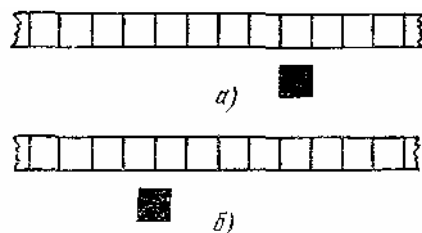
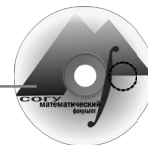


Рис. 4. Когда каретка неподвижна, она стоит против одной из секций ленты так, как показано на рис. а), а не так, как показано на рис. б). Ситуация, изображенная на рис. б), может возникнуть только в процессе движения каретки.

Каретка может передвигаться вдоль ленты влево и вправо. Когда она неподвижна, она стоит против ровно одной секции ленты (рис. 4, а; на этом и следующих чертежах каретка изображена и виде зачерненного квадрата); говорят, что каретка *обозревает* эту секцию, или держит ее в *поле зрения*.

Информация о том, какие секции пусты, а какие отмечены и где стоит каретка, образует *состояние машины Поста*. Таким образом, состояние машины складывается из состояния ленты и указания номера той секции, которую обозревает каретка. За единицу времени {которую мы будем называть *шагом*) каретка может сдвинуться на одну секцию влево или вправо. Кроме того, каретка может поставить (напечатать) или уничтожить (стереть) метку в той секции, против которой она стоит, а также распознать, стоит или нет метка в обозреваемой ею секции.

Программа машины Поста

Работа машины Поста состоит в том, что каретка передвигается вдоль ленты и печатает или стирает метки. Эта работа происходит по инструкции определенного вида, называемой программой. Для машины Поста возможно составление различных программ. Посмотрим, как устроена программа.

Каждая программа машины Поста состоит из команд. *Командой машины Поста* будем называть выражение, имеющее один из следующих шести видов (буквы i , j , j_1 , j_2 означают всюду натуральные числа $1, 2, 3, 4, 5, \dots$):

Первый вид. Команды движения вправо

$$i. \Rightarrow j$$



Второй вид. Команды движения влево

$$i. \Leftarrow j$$

Третий вид. Команды печатания метки

$$i. \vee j$$

Четвертый вид. Команды стирания метки

$$i. \xi j$$

Пятый вид. Команды передачи управления

$$i. ? \begin{cases} j_1 \\ j_2 \end{cases}$$

Шестой вид. Команды остановки

$$i. \text{ стоп.}$$

Например,

$$137. \Rightarrow 1$$

является командой движения вправо,

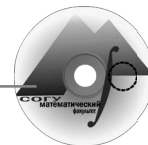
$$25. ? \begin{cases} 32 \\ 25 \end{cases}$$

— командой передачи управления, а

$$6386. \text{ стоп}$$

— командой остановки.

Число 1, стоящее в начале команды, называется *номером* команды. Так, у приведенных только что команд номера суть соответственно 137, 25 и 6386. Число j ,



стоящее в конце команды (а у команд передачи управления — каждое из чисел j_1 и j_2), будем называть *отсылкой* (при этом в команде передачи управления j_1 — верхней, а j_2 — нижней отсылкой). У команд остановки нет отсылки. Так, у приведенных только что команд отсылками служат числа 1, 32, 25, причем 32 — верхняя отсылка, а 25 — нижняя отсылка.

Программой машины Поста будем называть конечный непустой (т. е. содержащий хотя бы одну команду) список команд машины Поста, обладающий следующими двумя свойствами:

1. На первом месте в этом списке стоит команда с номером 1, на втором месте (если оно есть) — команда с номером 2 и т. д.; вообще на A -м месте стоит команда с номером A .
2. Отсылка любой из команд списка совпадает с номером некоторой (другой или той же самой) команды списка (более точно: для каждой отсылки каждой команды списка найдется в списке такая команда, номер которой равен рассматриваемой отсылке).

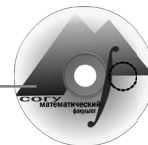
Например, следующий список будет программой машины Поста:

1. стоп	3. § 3
2. ? $\begin{cases} \nearrow 4 \\ \searrow 1 \end{cases}$	4. стоп.

А эти два списка не будут программами машины Поста, хотя и составлены из команд машины Поста:

2. ? $\begin{cases} \nearrow 4 \\ \searrow 1 \end{cases}$	3. § 3
1. стоп	4. стоп

(не выполнено первое условие);



1. стоп 3. ξ 3
2. ? $\begin{cases} 4 \\ 5 \end{cases}$ 4. стоп

(не выполнено второе условие).

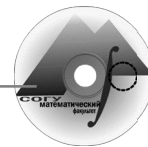
Для наглядности программы машины Поста мы будем записывать столбиком. Число команд программы называется длиной программы.

Упражнение. Выпишите все программы машины Поста длины 1. Сколько существует программ. длины 2, длины 3, длины n ?

Работа машины Поста

Чтобы машина Поста начала работать, надо задать, во-первых, некоторую программу, а во-вторых, некоторое ее (машины) состояние, т. е. как-то расставить метки по секциям ленты (в частности, можно все секции оставить пустыми) и поставить каретку против одной из секций. Как правило, мы будем предполагать, что в начальном (т. е. в задаваемом вначале) состоянии машины каретка ставится всегда против секции с номером (координатой) нуль. При таком соглашении начальное состояние машины полностью определено состоянием ленты.

Как уже говорилось, программа является той инструкцией, на основании которой работает машина. Работа машины на основании заданной программы (и при заданном начальном состоянии) происходит следующим образом. Машина приводится в начальное состояние и приступает к выполнению первой команды программы (что значит «выполнить команду», будет объяснено ниже). Эта команда выполняется за один шаг, после чего машина приступает к выполнению той команды, номер которой (назовем его k) равен отсылке (одной из отсылок, если их две) первой команды. Эта команда также выполняется за один шаг, после чего начинается выполнение команды, номер которой равен отсылке команды с номером a . Вообще каждая команда выполняется за один шаг, а переход от выполнения одной команды к выполнению другой происходит по следующему правилу: пусть на k -м шаге выполнялась команда с номером i , тогда, если эта команда имеет единственную отсылку j , то на $k+1$ -м шаге выполняется команда с номером j ; если эта команда имеет две отсылки j_1 и j_2 , то на k

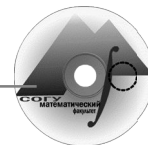


+ 1-м шаге выполняется одна из двух команд— с номером j_1 или с номером j_2 (какая именно, будет указано ниже); если, наконец, выполняющаяся на k -м шаге команда вовсе не имеет отсылки, то на $k+1$ -м шаге и на всех последующих шагах не выполняется никакая команда: машина останавливается. Осталось объяснить, что значит выполнить команду и какая из отсылок — при наличии двух — выбирается в качестве номера следующей команды.

Выполнение команды движения вправо состоит в том, что каретка сдвигается на одну секцию вправо. Выполнение команды движения влево состоит в том, что каретка сдвигается на одну секцию влево. Выполнение команды печатания метки состоит в том, что каретка ставит метку на обозреваемой секции; выполнение этой команды возможно лишь в том случае, если обозреваемая перед началом выполнения команды секция пуста; если же на обозреваемой, секции уже стоит метка, команда считается невыполнимой. Выполнение команды стирания метки состоит в том, что каретка уничтожает метку в обозреваемой секции; выполнение этой команды возможно лишь в том случае, если обозреваемая секция отмечена; если же на обозреваемой секции и так нет метки, команда считается невыполнимой. Выполнение команды передачи управления с верхней отсылкой j_1 и нижней отсылкой j_2 никак не изменяет состояния машины: ни одна из меток не уничтожается и не ставится, и каретка также остается неподвижной (машина делает, так сказать, «шаг на месте»); однако если секция, обозреваемая перед началом выполнения этой команды, была пуста, то следующей должна выполняться команда с номером j_1 , если же эта секция была отмечена, следующей должна выполняться команда с номером j_2 (роль команды передачи управления сводится, следовательно, к тому, что каретка во время выполнения этой команды как бы «распознает», стоит ли перед ней метка, — именно это имелось в виду в предпоследней фразе § 1). Выполнение команды остановки тоже никак не меняет состояния машины и состоит в том, что машина останавливается.

Если теперь, задавшись программой и каким-либо начальным состоянием, пустить машину в ход, то осуществится один из следующих трех вариантов:

1) В ходе выполнения программы машина дойдет до выполнения невыполнимой команды {печатания метки в непустой секции или стирания метки в пустой секции); выполнение программы тогда прекращается, машина



останавливается; происходит так называемая *безрезультатная остановка*.

2) В ходе выполнения программы машина дойдет до выполнения команды остановки; программа в этом случае считается выполненной, машина останавливается; происходит так называемая *результативная остановка*.

3) В ходе выполнения программы машина не дойдет до выполнения ни одной из команд, указанных в первых двух вариантах; выполнение программы при этом никогда не прекращается, машина никогда не останавливается; процесс работы машины происходит бесконечно.

Примеры выполнения программ

Зададим, например, начальное состояние, указанное на рис. 5, и следующую программу:

- | | |
|-------------------|---------------------------------------|
| 1. $\vee 4$ | 4. $\Rightarrow 5$ |
| 2. $\xi 3$ | 5. $\begin{cases} 4 \\ 3 \end{cases}$ |
| 3. $\Leftarrow 2$ | |

Посмотрим, как будет работать машина при таком начальном состоянии и такой программе.

На первом шаге будет выполняться команда № 1. После первого шага состояние машины станет таким,



Рис. 5.



Рис. 6.

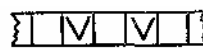


Рис. 7.

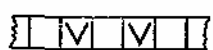


Рис. 8.

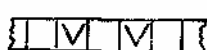


Рис. 9.

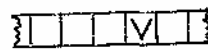
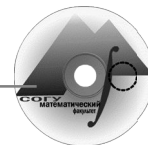


Рис. 10.

как указано на рис. 6. После выполнения команды № 1 надо перейти к выполнению той команды, номер которой совпадает с отсылкой команды № 1, т. е. к выполнению команды № 4. Эта команда будет выполняться в течение второго шага,



и состояние машины станет таким, как на рис. 7. Теперь надо выполнить команду № 5 (ибо отсылка команды № 4 равна 5). Эта команда будет выполняться на третьем шаге, в результате которого состояние машины не изменится и останется таким, как на рис. 7. Поскольку обозреваемая секция при этом пуста, то следующей должна выполняться команда, номер которой равен верхней отсылке, т. е. числу 4. После выполнения на четвертом шаге команды № 4 машина придет в состояние, указанное на рис. 8. Теперь, на пятом шаге, будет выполняться команда № 5. На этот раз обозреваемая секция отмечена, поэтому следующей будет выполняться команда с номером, равным нижней отсылке, т. е. числу 3. После выполнения на шестом шаге команды № 3 машина придет в состояние, показанное на рис. 9, и приступит на седьмом шаге к выполнению команды № 2. Однако команда № 2 окажется невыполнимой, поскольку предписывает стереть метку в пустой секции. Следовательно, на седьмом шаге произойдет безрезультатная остановка.

Различные программы, примененные к одному и тому же начальному состоянию, могут привести к различным исходам: безрезультатной остановке, результативной остановке, безостановочной работе машины. Действительно, зададимся, например, начальным состоянием, указанным на рис. 10. Применим к ЭТОМУ начальному состоянию программу:

1. $\Rightarrow 2$
2. $\Rightarrow 3$
3. $\vee$.

Машина сделает два шага, а на третьем шаге произойдет безрезультатная остановка. Применим к этому начальному состоянию программу:

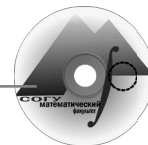
1. $\Rightarrow 2$
2. $\Rightarrow 3$
3. стоп.

Машина сделает два шага, а затем на третьем произойдет результативная остановка.

1. $\Rightarrow 1$.

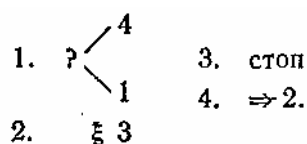
Машина будет работать бесконечно.

1. $\begin{cases} 1 \\ ? \\ 1 \end{cases}$.



Применим, наконец, к этому же начальному состоянию программу:

Машина опять-таки будет работать бесконечно (несмотря на то, что ни запись на ленте, ни положение каретки не будут при этом меняться). Точно так же различные варианты может давать и одна и та же программа, примененная к различным начальным состояниям. Рассмотрим, например, следующую программу:



Применим ее к начальным состояниям А, Б, В, показанным на рис. 11. Для начального состояния А получаем результативную остановку на четвертом шаге,

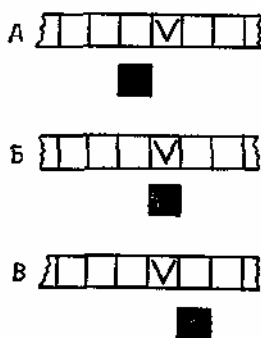
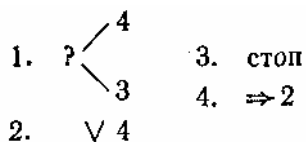


Рис. 11.

для Б — бесконечную работу машины, для В — безрезультатную остановку на третьем шаге. Применение к тем же начальным состояниям программы



дает безрезультатную остановку для А, результативную остановку для Б и бесконечную работу машины для В.

Упражнение 1. Может ли быть программа, дающая при любом начальном состоянии результативную остановку? Безрезультатную остановку? Неограниченное продолжение работы машины? Каково наименьшее число команд в этих программах?



Упражнение 2. О некоторой программе известно, что существует начальное состояние, для которого она даёт результативную остановку, и существует начальное состояние, для которого она даёт непрекращающуюся работу машины. Докажите, что число команд в этой программе не менее 4. напишите все такие программы длины 4.