

**НОРМАТИВНЫЙ КОНСПЕКТ ПО ДИСЦИПЛИНЕ
«ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ В ГУМАНИТАРНОЙ СФЕРЕ»**

Часть 3

**«АЛГОРИТМИЧЕСКАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ.
КОНСТРУИРОВАНИЕ МАШИН ТЬЮРИНГА»**

Введение. Описанное нами «устройство» машины Тьюринга (разбитая на ячейки лента, считывающе-записывающая головка) по существу не имеет никакого значения. Машина Тьюринга есть не что иное, как некоторая эффективная процедура (алгоритм) для преобразования слов алфавита $\{a_0, a_1, a_2, \dots, a_n, q_0, q_1, q_2, \dots, q_n\}$, то есть конфигураций. Таким образом, для определения машины Тьюринга нужно задать ее внешний $\{a_0, a_1, a_2, \dots, a_n\}$ и внутренний $\{q_0, q_1, q_2, \dots, q_n\}$ алфавиты, функциональную схему (программу) и указать, какие из символов обозначают пустую ячейку, начальное и заключительное состояние.

Применение машины Тьюринга к слову. Пусть дана машина Тьюринга с внешним алфавитом $A = \{a_0, I\}$ (здесь a_0 есть символ пустой ячейки), алфавитом внутренних состояний $Q = \{q_0, q_1, q_2\}$ и со следующей функциональной схемой (программой):

$$q_1 a_0 \rightarrow q_2 a_0 R \quad (R_1)$$

$$q_2 a_0 \rightarrow q_0 I \quad (R_2)$$

$$q_1 I \rightarrow q_1 I R \quad (R_3)$$

$$q_2 I \rightarrow q_2 I R \quad (R_4)$$

В исходной конфигурации машина находится во внутреннем состоянии q_1 , головка обозревает крайнюю правую непустую ячейку. Требуется определить, в какое слово переработает эта машина слово $I a_0 I$.

Будем последовательно выписывать конфигурации машины, возникающие при переработке ею этого слова. Имеем начальную конфигурацию C_1 :

$$\begin{array}{c} q_1 \\ I \ a_0 \ I \end{array} \quad (C_1)$$

На первом такте работы действует команда R_3 , ибо машина во внутреннем состоянии q_1 обозревает ячейку с символом I . В обозреваемую ячейку записывается символ I (то есть в ней ничто не меняется), а головка смещается на одну позицию вправо (там «нет ничего», то есть есть символ пустой ячейки). Внутреннее состояние машины меняется с q_1 на q_1 (то есть остается прежним и равным q_1). В результате создается следующая конфигурация C_2 :

$$\begin{array}{c} q_1 \\ I \ a_0 \ I \ a_0 \end{array} \quad (C_2)$$

Машина во внутреннем состоянии q_1 обозревает пустую ячейку. Срабатывает команда R_1 . Внутреннее состояние меняется на q_2 , в обозреваемую ячейку вписывается символ пустой ячейки (то есть в ней ничего не меняется), головка смещается на одну позицию вправо (там «нет ничего», то есть есть символ пустой ячейки). По-

лучается такая конфигурация C_3 :

$$I \ a_0 \ I \ a_0 \ a_0 \quad (C_3) \quad q_2$$

Машина во внутреннем состоянии q_2 обозревает пустую ячейку. Срабатывает команда R_2 . Внутреннее состояние меняется на q_0 , в обозреваемую ячейку записывается символ I , головка остается на месте. Получается такая конфигурация C_4 :

$$I \ a_0 \ I \ a_0 \ I \quad (C_4) \quad q_0$$

Машина во внутреннем состоянии q_0 (а это завершающее состояние) обозревает ячейку с символом I . Ни одна команда не имеет подходящей левой части. Ни одна команда не срабатывает. Машина останавливается (то есть конфигурация C_4 является завершающей). Если обозначить рассмотренную нами машину Тьюринга символом Θ , то можно записать: $\Theta(1a_0I) = 1a_0Ia_0I$, что означает: машина Θ оказалась применимой к слову $1a_0I$ и переработала это входное слово в выходное слово $1a_0Ia_0I$.

Использованный нами способ записи конфигураций машины Тьюринга (*над* обозреваемой ячейкой записывается имя внутреннего состояния) удобен для рукописной нотации. Такой способ использовался самим Тьюрингом, и его называют *нотацией Тьюринга*. Можно также записывать имя внутреннего состояния *перед* обозреваемой ячейкой. Этот способ удобен для типографской нотации, его называют *нотацией Поста* (в память выдающегося американского математика и логика Эмиля Леона Поста, в 1936 году немного позже Тьюринга предложившего свою формализацию понятия «алгоритм» – машину Поста). Используя нотацию Поста, процесс и результат переработки машиной Θ слова $1a_0I$ можно записать так:

$$\Theta(1a_0I) = 1a_0q_1I \rightarrow R_3 \rightarrow 1a_0Iq_1a_0 \rightarrow R_1 \rightarrow 1a_0Ia_0q_2a_0 \rightarrow R_2 \rightarrow 1a_0Ia_0q_0I.$$

Конструирование машины Тьюринга: пример с перемещением головки и заменой символов. Пусть входное слово P есть непустое слово в алфавите $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Иными словами, P есть последовательность из десятичных цифр, то есть запись неотрицательного целого числа в десятичной системе счисления. В качестве выходного слова требуется получить на ленте запись числа, которое на 1 больше числа, представленного входным словом P .

Для решения этой задачи можно выполнить следующие действия (неформальное описание алгоритма решения задачи):

1. Перегнать головку к последней цифре.
2. Если это цифра от 0 до 8 , то заменить ее цифрой на 1 больше и остановиться. Например, входному слову 1957 будет соответствовать выходное слово 1958 .
3. Если последняя цифра есть 9 , тогда заменить ее на 0 и сдвинуть головку влево к предыдущей цифре, после чего увеличить на 1 эту предпоследнюю цифру. Например, входному слову 649 будет соответствовать выходное слово 650 . Если предпоследняя цифра тоже была 9 , то она заменится тоже нулем, а ей предшествующая цифра увеличится на единицу и так далее. Если входное слово состоит не из

одних только девяток, то найдется та из предшествующих цифр, которая увеличится на 1, но левее ее изменений уже не будет.

4. Если в P имеются только девятки (как в слове, например, 99), тогда головка будет сдвигаться влево, заменяя девятки на нули, и в конце концов окажется над пустой ячейкой. В эту пустую ячейку надо записать 1 и остановиться (для входного слова 99 выходным словом будет 100).

Функциональная схема (программа) для машины Тьюринга, решающей эту задачу, может быть такова (формальное описание алгоритма решения задачи):

$q_10 \rightarrow q_10R$	$q_20 \rightarrow q_01$
$q_11 \rightarrow q_11R$	$q_21 \rightarrow q_02$
$q_12 \rightarrow q_12R$	$q_22 \rightarrow q_03$
$q_13 \rightarrow q_13R$	$q_23 \rightarrow q_04$
$q_14 \rightarrow q_14R$	$q_24 \rightarrow q_05$
$q_15 \rightarrow q_15R$	$q_25 \rightarrow q_06$
$q_16 \rightarrow q_16R$	$q_26 \rightarrow q_07$
$q_17 \rightarrow q_17R$	$q_27 \rightarrow q_08$
$q_18 \rightarrow q_18R$	$q_28 \rightarrow q_09$
$q_19 \rightarrow q_19R$	$q_29 \rightarrow q_20L$
$q_1a_0 \rightarrow q_2a_0L$	$q_2a_0 \rightarrow q_01$

Смысл имен внутренних состояний:

- q_1 – это внутреннее состояние, в котором головка «бежит» к последней цифре числа. Для этого она все время движется вправо, не меняя обозреваемые символы и оставаясь в том же внутреннем состоянии. Но здесь есть одна особенность: когда головка находится над последней цифрой, то она еще не знает об этом (ведь она не видит того, что записано в ячейках, соседних с обозреваемой) и определит это лишь тогда, когда попадет на пустую ячейку. Поэтому, дойдя до первой пустой ячейки, головка возвращается назад к последней цифре и переходит во внутреннее состояние q_2 (вправо двигаться уже не нужно);

- q_2 – это состояние, в котором машина прибавляет 1 к той цифре, которую видит в данный момент. Сначала это последняя цифра числа; если она принадлежит диапазону $0..8$, то машина заменяет ее цифрой, которая на 1 больше, и останавливается. Но если это цифра 9, то машина заменяет ее на 0 и сдвигает головку влево, оставаясь в состоянии q_2 . Значит, теперь будет прибавляться 1 к предыдущей цифре. Если и эта цифра равна 9, то машина заменяет ее на 0 и сдвигает головку влево, оставаясь по-прежнему в состоянии q_2 , так как нужно выполнить то же самое действие – увеличить на 1 обозреваемую цифру. Если же головка сдвинулась влево, а в обозреваемой ячейке нет цифры (ячейка пуста), то машина в нее записывает 1 и останавливается.

Конструирование машины Тьюринга: пример с запоминанием символа. Входным является слово P в алфавите $A = \{a, b, c\}$. Если слово P не является пустым, то машина Тьюринга Γ должна перенести его первый символ в конец. Например, $\Gamma(bac) = acb$.

Для решения этой задачи можно выполнить следующие действия:

1. Запомнить первый символ входного слова P , а затем стереть этот символ.
2. Перегнать головку вправо под первую пустую ячейку за P и записать в нее запомненный символ.

Осуществление прогона вправо не составляет проблемы. А как запомнить первый символ? Ведь у машины Тьюринга (на первый взгляд) нет другого запоминающего устройства, кроме ленты. Но запоминать символ в какой-то ячейке на ленте бессмысленно: как только головка сдвинется влево или вправо от этой ячейки, она тут же забудет данный символ. Выход здесь таков – надо запоминать информацию в именах внутренних состояний. Заметим вместе с тем, что вовсе не обязательно использовать в качестве имен внутренних состояний $q_1, q_2, q_3, q_4, \dots, q_n, q_0$. Ведь внутренний алфавит Q машины Тьюринга, как и любой алфавит, есть произвольное конечное множество. А значит, его элементами (в его смысле символами) вполне могут быть объекты, которые в смысле других алфавитов являются целыми словами. Проще говоря, вполне можно использовать и такие имена внутренних состояний, как, например *go_right*, *end*, *здесь_мы_видели_0* и другие им подобные «говорящие имена». Договоримся только, что именем начального внутреннего состояния всегда будет q_1 , а именем завершающего внутреннего состояния – q_0 . Введенный нами способ именования внутренних состояний машины Тьюринга будем называть *расширенной нотацией* этих состояний.

Итак, если первый символ входного слова есть a , то надо перейти во внутреннее состояние *memento- a* , в котором головка бежит вправо и записывает в конце a . Если же первым был символ b , тогда надо перейти во внутреннее состояние *memento- b* , где делается все то же самое, только в конце записывается символ b . Если же первым был символ c , тогда переходим в состояние *memento- c* , в котором дописывается за входным словом символ c . Следовательно, то, каким был первый символ, мы фиксируем переводом машины в разные внутренние состояния. Этот прием является типичным при конструировании машин Тьюринга. Функциональная схема (программа) конструируемой машины Тьюринга Γ будет такой:

$$q_1 a_0 \rightarrow q_0 a_0$$

Если входное слово является пустым, то сразу останавливаемся.

$$q_1 a \rightarrow \textit{memento-}a a_0 R$$

Запоминаем и стираем первый символ.

$$q_1 b \rightarrow \textit{memento-}b a_0 R$$

$$q_1 c \rightarrow \textit{memento-}c a_0 R$$

$$\textit{memento-}a a \rightarrow \textit{memento-}a a R$$

$$\textit{memento-}a b \rightarrow \textit{memento-}a b R$$

$$\textit{memento-}a c \rightarrow \textit{memento-}a c R$$

$$\textit{memento-}b a \rightarrow \textit{memento-}b a R$$

$$\textit{memento-}b b \rightarrow \textit{memento-}b b R$$

$$\textit{memento-}b c \rightarrow \textit{memento-}b c R$$

$$\textit{memento-}c a \rightarrow \textit{memento-}c a R$$

*В одном из трех внутренних состояний (*memento- a* , *memento- b* или *memento- c*) двинемся к первой справа пустой ячейке.*

$memento-c\ b \rightarrow memento-c\ bR$
 $memento-c\ c \rightarrow memento-c\ cR$

$memento-a\ a_0 \rightarrow q_0a$
 $memento-b\ a_0 \rightarrow q_0b$
 $memento-c\ a_0 \rightarrow q_0c$

*Впечатываем хранимый в памяти
внутреннего состояния символ
и останавливаемся.*

Применяя эту машину к различным входным словам, можно убедиться, что она корректно перерабатывает и пустое слово, и слова, содержание более одной буквы. Если во входном слове ровно один символ, тогда машина сотрет этот символ, сдвинется на одну ячейку вправо и запишет в нее данный символ. Таким образом, однобуквенное слово попросту сдвинется на одну ячейку вправо. Это допустимо. Ведь ячейки ленты не нумерованы, поэтому местоположение слова на ленте никак не фиксируется и перемещение слова влево или вправо «заметить нельзя». В связи с этим не требуется, чтобы выходное слово обязательно находилось в том же месте, где было входное слово. Результат может оказаться как левее, так и правее этого места.

Конструирование машины Тьюринга: пример со сравнением символов и стиранием слова. Пусть входное слово P есть слово в алфавите $A = \{a, b, c\}$. Если это слово не является пустым, и его первый и последний символы одинаковы, тогда машина Тьюринга Ψ не должна это слово менять, а иначе должна заменить его на пустое слово.

Для решения этой задачи можно выполнить следующие действия:

1. Запомнить первый символ входного слова, не стирая его.
2. Переместить головку к последнему символу и сравнить его с запомненным.

Если они одинаковы, то больше ничего не делать.

3. В противном случае уничтожить все входное слово.

Как запоминать символ и как перегонять головку к последнему символу слова известно из предыдущих примеров. Стирание же входного слова реализуется заменой всех его символов на символ a_0 . При этом, раз уж головка оказалась в конце слова, будем перемещать ее справа налево до первой пустой ячейки. Эти действия описываются следующей функциональной схемой (программой):

$q_1a_0 \rightarrow q_0a_0$

*Если входное слово является пустым,
то сразу завершаем работу.*

$q_1a \rightarrow mem-a\ aR$
 $q_1b \rightarrow mem-b\ bR$
 $q_1c \rightarrow mem-c\ cR$

Запоминаем первый символ.

$mem-a\ a \rightarrow mem-a\ aR$
 $mem-a\ b \rightarrow mem-a\ bR$
 $mem-a\ c \rightarrow mem-a\ cR$
 $mem-b\ a \rightarrow mem-b\ aR$
 $mem-b\ b \rightarrow mem-b\ bR$

В одном из трех внутренних состояний ($mem-a$, $mem-b$ или $mem-c$) движемся к первой справа пустой ячейке.

$mem-b\ c \rightarrow mem-b\ cR$
 $mem-c\ a \rightarrow mem-c\ aR$
 $mem-c\ b \rightarrow mem-c\ bR$
 $mem-c\ c \rightarrow mem-c\ cR$

$mem-a\ a_0 \rightarrow compare-a\ a_0L$ Достигнув пустой ячейки,
 $mem-b\ a_0 \rightarrow compare-b\ a_0L$ возвращаемся на одну позицию
 $mem-c\ a_0 \rightarrow compare-c\ a_0L$ влево с целью произвести
 сравнение символов.

$compare-a\ a \rightarrow q_0a$ Если символы совпадают,
 $compare-b\ b \rightarrow q_0b$ то работа завершается.
 $compare-c\ c \rightarrow q_0c$

$compare-a\ b \rightarrow erase\ a_0L$ Если символы не совпадают,
 $compare-a\ c \rightarrow erase\ a_0L$ то стираем последний символ
 $compare-b\ a \rightarrow erase\ a_0L$ и переходим во "всестирающее"
 $compare-b\ c \rightarrow erase\ a_0L$ внутреннее состояние $erase$.
 $compare-c\ a \rightarrow erase\ a_0L$
 $compare-c\ b \rightarrow erase\ a_0L$

$erase\ a \rightarrow erase\ a_0L$ Во внутреннем состоянии $erase$
 $erase\ b \rightarrow erase\ a_0L$ движемся влево, стирая все
 $erase\ c \rightarrow erase\ a_0L$ встречающиеся непобельные
 символы.

$erase\ a_0 \rightarrow q_0a_0$ Когда все слово стерто,
 завершаем работу.

Конструирование машины Тьюринга: пример с удалением символа из слова. Входным является слово P в алфавите $A = \{a, b\}$. Машина Тьюринга Σ должна удалить из слова P его второй символ, если такой имеется.

Казалось бы, эту задачу решить просто: надо сдвинуть головку к ячейке со вторым символом и затем очистить эту ячейку. Однако, внутри выходного слова не должно быть пустых ячеек. Поэтому, после удаления второго символа надо «сжать» слово, перенести первый символ на одну позицию вправо. Для этого головка должна вернуться к первому символу, запомнить его и стереть, а затем, снова сдвинувшись вправо, записать запомненный символ в ту ячейку, где был второй символ. Алгоритм можно оптимизировать: начальный «поход» вправо ко второму символу, чтобы его стереть, и последующий возврат к первому символу являются лишними действиями. Можно сразу запомнить первый символ, стереть его и записать вместо второго символа. Функциональная схема (программа) машины Тьюринга Σ будет такова:

$$q_1 a_0 \rightarrow q_0 a_0$$

*Если входное слово является пустым,
то сразу останавливаемся.*

$$q_1 a_0 \rightarrow \text{mem-}a \ a_0 R$$

$$q_1 b \rightarrow \text{mem-}b \ a_0 R$$

*Запоминаем и стираем первый символ.
Сдвигаемся ко второй ячейке.*

$$\text{mem-}a \ a \rightarrow q_0 a$$

$$\text{mem-}a \ b \rightarrow q_0 a$$

$$\text{mem-}a \ a_0 \rightarrow q_0 a$$

$$\text{mem-}b \ a \rightarrow q_0 b$$

$$\text{mem-}b \ b \rightarrow q_0 b$$

$$\text{mem-}b \ a_0 \rightarrow q_0 b$$

*Впечатываем запомненный символ
и останавливаемся.*

Конструирование машины Тьюринга: пример со сжатием слова. Входное слово P есть слово в алфавите $A = \{a, b, c\}$. Машина Тьюринга Ω должна удалить из слова P первое вхождение символа a , если оно имеется.

Для решения этой задачи можно выполнить следующие действия:

1. Если в исходном внутреннем состоянии q_1 машина обозревает пустую ячейку, значит входное слово является пустым. В нем нет вхождений символа a . Необходимо просто завершить работу.

2. Если в исходном внутреннем состоянии q_1 машина обозревает ячейку с символом b или c , то надо, не меняя внутреннего состояния, двигаться вправо. Если во входном слове нет вхождений символа a , то головка придет к пустой ячейке. Машину здесь нужно остановить.

3. Если, двигаясь вправо во внутреннем состоянии q_1 , машина придет к ячейке, содержащей символ a , то нужно стереть его и в новом внутреннем состоянии (назовем его $[?]$) сместить головку вправо, чтобы узнать, какой символ там находится.

4. Символ, находящийся справа, надо запомнить (перейдя в одно из трех новых внутренних состояний, назовем их $\{a\}$, $\{b\}$, $\{c\}$) и стереть. После этого сместить головку на одну позицию влево и в имеющуюся там пустую ячейку записать «перенесенный» справа символ. Теперь справа будет пустая ячейка, поэтому «проедем» ее во внутреннем состоянии $[?]$ и вернемся в состояние $[?]$.

5. Если будут встречены символы a , b или c , то повторятся действия, описанные в предыдущем пункте. Так (в цикле) пустота будет перемещаться вправо, а имеющиеся справа от пустоты символы будут перемещаться влево: слово сжимается, а пустота вытесняется «в конец».

6. Когда пустота будет окончательно вытеснена «в конец», машина, находясь во внутреннем состоянии $[?]$, встретит пустую ячейку. Здесь машину нужно остановить.

Функциональная схема (программа) машины Тьюринга Ω может быть такой:

$$q_1 a_0 \rightarrow q_0 a_0$$

*Если входное слово является пустым, то сразу
останавливаемся. Эта же команда остановит
машину, если во входном слове нет вхождений
искомого символа.*

$$q_1b \rightarrow q_1bR$$

$$q_1c \rightarrow q_1cR$$

$$q_1a \rightarrow [?]a_0R$$

В исходном внутреннем состоянии совершаем прогон через символы b и c . Встретив символ a , стираем его и во внутреннем состоянии $[?]$ смотрим, какой символ стоит справа от образовавшейся пустой ячейки.

$$[?]a \rightarrow \{a\}a_0L$$

$$[?]b \rightarrow \{b\}a_0L$$

$$[?]c \rightarrow \{c\}a_0L$$

$$\{a\}a_0 \rightarrow \{ \}aR$$

$$\{b\}a_0 \rightarrow \{ \}bR$$

$$\{c\}a_0 \rightarrow \{ \}cR$$

$$\{ \}a_0 \rightarrow [?]a_0R$$

Символы, стоящие справа от пустоты, перескакивают влево, пустота перескакивает вправо.

$$[?]a_0 \rightarrow q_0a_0L$$

Полностью вытеснив пустоту «в конец», останавливаемся.

Конструирование машины Тьюринга: пример со вставкой символа в слово. Входное слово P есть слово в алфавите $A = \{a, b, c\}$. Если слово P не является пустым, то машина Тьюринга Δ должна за его первым символом вставить символ a .

Решить эту задачу можно следующим образом:

1. Если в исходном внутреннем состоянии q_1 машина обозревает пустую ячейку, значит входное слово является пустым. В согласии с условием задачи необходимо просто завершить работу.

2. Если входное слово не является пустым, необходимо запомнить (путем перехода в одно из новых внутренних состояний $\{a\}$, $\{b\}$, $\{c\}$) и затереть его первый символ, заменив его на a . Головку следует сместить на одну позицию влево.

3. В обозреваемую теперь пустую ячейку нужно вписать запомненный символ и завершить работу.

Функциональная схема (программа) машины Тьюринга Δ может быть такой:

$$q_1a_0 \rightarrow q_0a_0$$

Если входное слово является пустым, то сразу завершаем работу.

$$q_1a \rightarrow \{a\}aL$$

$$q_1b \rightarrow \{b\}aL$$

$$q_1c \rightarrow \{c\}aL$$

Запоминаем первый символ слова, заменяем его на a , сдвигаемся на одну позицию влево.

$$\{a\}a_0 \rightarrow q_0a$$

$$\{b\}a_0 \rightarrow q_0b$$

$$\{c\}a_0 \rightarrow q_0c$$

Вписываем запомненный символ и завершаем работу.

Конструирование машины Тьюринга: пример с раздвижкой слова. Входное слово P есть слово в алфавите $A = \{a, b, c\}$. Машина Тьюринга Φ должна вставить в слово P символ a за первым вхождением символа c , если такое вхождение имеется.

Для решения этой задачи можно выполнить следующие действия:

1. Если в исходном внутреннем состоянии q_1 машина обозревает пустую ячейку, значит входное слово является пустым. В нем вхождений символа c нет. Необходимо просто завершить работу.

2. Если в исходном внутреннем состоянии q_1 машина обозревает ячейку с символом a или b , то надо, не меняя внутреннего состояния, двигаться вправо. Если во входном слове нет вхождений символа c , то головка придет к пустой ячейке. Машину здесь нужно остановить.

3. Если в обрабатываемом слове есть вхождение символа c , то головка придет к ячейке, содержащей этот символ. Не затирая его, переведем машину во внутреннее состояние $[?]$ и сдвинем головку на одну позицию вправо. Если там обнаружится пустая ячейка, значит первое вхождение символа c в обрабатываемом слове имеется в конце. Значит, находясь во внутреннем состоянии $[?]$ и обозревая пустую ячейку, надо записать в нее символ a и завершить работу.

4. Если работа не завершилась на предыдущем шаге, то, какой бы символ ни обнаружила машина, бывшая во внутреннем состоянии $[?]$, этот символ нужно запомнить (перейдя в одно из внутренних состояний $[a]$, $[b]$, $[c]$), впечатать вместо него символ a и сдвинуть головку вправо.

5. Обнаружив справа затем любой непробельный символ, его надо запомнить, впечатать вместо него хранимый в памяти к настоящему моменту символ и сдвинуть головку вправо. Так следует делать до тех пор, пока в одном из внутренних состояний $[a]$, $[b]$, $[c]$ машина не придет к пустой ячейке. В эту пустую ячейку нужно впечатать соответствующий внутреннему состоянию последний запомненный символ и завершить работу.

Функциональная схема (программа) машины Тьюринга Φ может быть такой:

$q_1 a_0 \rightarrow q_0 a_0$

Если входное слово является пустым, то сразу останавливаемся. Эта же команда остановит машину, если во входном слове нет вхождений символа c .

$q_1 a \rightarrow q_1 a R$

$q_1 b \rightarrow q_1 b R$

В исходном внутреннем состоянии q_1 совершаем прогон через ячейки, содержащие символы a и b .

$q_1 c \rightarrow [?] c R$

$[?] a_0 \rightarrow q_0 a$

Обнаружив ячейку с символом c , не затираем его, переходим во внутреннее состояние $[?]$ и смотрим вправо. Если там имеется пустая ячейка, вписываем в нее символ a и останавливаемся.

$[?] a \rightarrow [a] a R$

$[?] b \rightarrow [b] a R$

$[?] c \rightarrow [c] a R$

$[a] a \rightarrow [a] a R$

Если работа не завершена предыдущей командой, запоминаем обозреваемый символ, записываем вместо него символ a и сдвигаем головку вправо. Циклически запоминаем текущий символ, ввпечатываем

$[a\}b \rightarrow [b\}aR$
 $[a\}c \rightarrow [c\}aR$
 $[b\}a \rightarrow [a\}bR$
 $[b\}b \rightarrow [b\}bR$
 $[b\}c \rightarrow [c\}bR$
 $[c\}a \rightarrow [a\}cR$
 $[c\}b \rightarrow [b\}cR$
 $[c\}c \rightarrow [c\}cR$

ваем предыдущий запомненный символ и сдвигаем головку вправо.

$[a\}a_0 \rightarrow q_0a$
 $[b\}a_0 \rightarrow q_0b$
 $[c\}a_0 \rightarrow q_0c$

Когда текущий символ окажется символом пустой ячейки, впечатываем последний запомненный символ и останавливаемся.

Конструирование машины Тьюринга: пример с формированием выходного слова на новом месте. Входное слово P есть слово в алфавите $A = \{a, b, c\}$. Машина Тьюринга H должна удалить из слова P все вхождения символа a .

Предыдущие примеры показывают, что в функциональных схемах (программах) машин Тьюринга довольно сложно реализуются вставки символов в слова и удаления символов из слов. Поэтому иногда проще не раздвигать или сжимать входное слово, а формировать выходное слово в другом (свободном) месте ленты. Именно так мы и поступим при решении данной задачи. Предлагается выполнить следующие действия:

1. Находясь в исходном внутреннем состоянии q_1 , совершить прогон вправо, придя к первой пустой ячейке. В нее впечатать символ $=$, перейти во внутреннее состояние $\{ \}$. В этом внутреннем состоянии совершить прогон влево, дойдя до той пустой ячейки, которая предшествует началу обрабатываемого слова. Там перейти во внутреннее состояние $[? \}$, в обозреваемую пустую ячейку ничего не вписывать, сдвинуть головку вправо. Выполнив эти действия, машина припишет символ $=$ к концу обрабатываемого слова и во внутреннем состоянии $[? \}$ будет обозревать первый символ обрабатываемого слова.

2. Если машина, находясь во внутреннем состоянии $[? \}$, обозревает ячейку с символом a , то этот символ стирается, головка сдвигается на одну позицию вправо, внутреннее состояние не меняется. Если машина, находясь во внутреннем состоянии $[? \}$, обозревает ячейку с символом b или c , то этот символ запоминается (путем перехода во внутреннее состояние $[b\}$ или $[c\}$ соответственно) и стирается, головка смещается вправо с целью «сбегать» к концу слова (даже за знак $=$) и вписать запомненный символ в первую встреченную там пустую ячейку.

3. Во внутреннем состоянии $[b\}$ или $[c\}$ головка движется вправо, пока не достигнет пустой ячейки. В пустую ячейку записывается соответственно символ b или c , после чего машина, находясь в состоянии $\{ \}$, «гонит» головку влево до встречи с пустой ячейкой. Там переходит во внутреннее состояние $[? \}$, в обозреваемую пустую ячейку ничего не вписывает, сдвигает головку вправо. Если здесь обнаружится символ a , b или c , то действия, описанные в пунктах 2 и 3 будут циклически

повторяться, пока все символы a не будут затерты, а все символы b и c не «переко-чуют» с сохранением порядка их следования вправо за знак $=$.

4. Когда случится сказанное в конце пункта 3, машина во внутреннем состоянии $[?]$ «натолкнется» на символ $=$. Этот символ нужно затереть и завершить работу.

Функциональная схема (программа) машины Тьюринга H может быть такой:

$$q_1 a \rightarrow q_1 a R$$

$$q_1 b \rightarrow q_1 b R$$

$$q_1 c \rightarrow q_1 c R$$

$$q_1 a_0 \rightarrow \{ \} = L$$

$$\{ \} a \rightarrow \{ \} a L$$

$$\{ \} b \rightarrow \{ \} b L$$

$$\{ \} c \rightarrow \{ \} c L$$

$$\{ \} a_0 \rightarrow [?] a_0 R$$

Приписываем символ $=$ к концу слова. Возвращаемся к началу слова. Обозреваем первый символ слова, находясь во внутреннем состоянии $[?]$.

$$[?] a \rightarrow [?] a_0 R$$

$$[?] b \rightarrow [b] a_0 R$$

$$[?] c \rightarrow [c] a_0 R$$

Символ a просто затираем и переходим к следующему символу. Символ b или c запоминаем и стираем.

$$[b] a \rightarrow [b] a R$$

$$[b] b \rightarrow [b] b R$$

$$[b] c \rightarrow [b] c R$$

$$[b] = \rightarrow [b] = R$$

$$[b] a_0 \rightarrow \{ \} b L$$

$$[c] a \rightarrow [c] a R$$

$$[c] b \rightarrow [c] b R$$

$$[c] c \rightarrow [c] c R$$

$$[c] = \rightarrow [c] = R$$

$$[c] a_0 \rightarrow \{ \} c L$$

$$\{ \} = \rightarrow \{ \} = L$$

Относим запомненный символ в конец слова, формируемого справа от знака $=$, возвращаемся назад. Вернувшись, переходим к новому витку цикла, начиная его во внутреннем состоянии $[?]$.

$$[?] = \rightarrow q_0 a_0 R$$

Если все символы b и c уже перенесены вправо за знак $=$ (а все символы a затерты), то завершаем работу.

Конструирование машины Тьюринга: пример с фиксированием места на ленте. Входное слово P есть слово в алфавите $A = \{a, b\}$. Машина Тьюринга E должна удвоить (скопировать) слово P , поставив между ним и его копией знак $=$.

Эта задача решается аналогично предыдущей: в конец обрабатываемого слова записываем знак $=$, затем возвращаемся к началу слова и в цикле все его символы копируем в пустые ячейки справа от знака $=$.

Однако, имеется и отличие: копируемые символы обрабатываемого слова не удаляются, и это приводит к следующей проблеме. Записав справа копию очередного символа, мы затем должны вернуться к входному слову в позицию этого сим-

вола и потом сдвинуться вправо к следующему символу, чтобы скопировать уже его. Но как узнать, в какую позицию входного слова надо вернуться? В предыдущей задаче мы всегда возвращались к первому из оставшихся символов входного слова, а теперь мы сохраняем все символы, поэтому непонятно, какие символы мы уже скопировали, а какие еще нет. Отметим также, что в машине Тьюринга ячейки ленты никак не нумеруются, нет в машине Тьюринга и счетчиков, которые позволили бы определить, сколько символов мы уже скопировали.

В общем виде проблема, с которой мы столкнулись, следующая: как зафиксировать на ленте некоторую позицию, в которой мы уже были и к которой позже должны вернуться? Обычно эта проблема решается так. Когда мы оказываемся в этой позиции в первый раз, то заменяем находящийся в ней символ на его двойник – на новый вспомогательный символ. При этом разные символы заменяем на разные двойники, например a на A и b на B . После этого мы выполняем какие-то действия в других местах ленты. Чтобы затем вернуться к нашей позиции, надо просто отыскать на ленте ту ячейку, где находится символ-двойник. Затем в этой ячейке можно восстановить прежний символ, если нам больше не надо фиксировать эту позицию (именно для восстановления прежнего символа и надо было заменять разные символы на разные двойники).

Воспользуемся этим приемом в нашей задаче, выполняя следующие действия:

1. Во внутреннем состоянии q_1 совершаем прогон вправо, в первую справа пустую ячейку вписываем символ $=$. Изменив внутреннее состояние на $\{ \}$, возвращаемся влево, пока не встретим первую слева пустую ячейку. Такой возврат впоследствии, возможно, будет повторяться. В этом случае идти влево во внутреннем состоянии $\{ \}$ будем до ячейки с текущим символом-двойником A или B .

2. Если, идя влево так, как описано в предыдущем пункте, мы дошли до пустой ячейки, то ничего в нее не вписываем и делаем сдвиг вправо на одну позицию, изменив внутреннее состояние на $[?]$. Если же в результате «похода влево» мы дошли до ячейки с символом-двойником, то вписываем в нее на место двойника исходный символ (на место A вписываем a , на место B вписываем b) и делаем сдвиг вправо на одну позицию, изменив внутреннее состояние на $[?]$.

3. Запоминаем символ (a или b), который встретится нам во внутреннем состоянии $[?]$, путем перехода во внутреннее состояние $[a]$ или $[b]$ соответственно. Сам этот символ заменяем на его двойник (A или B соответственно), головку сдвигаем вправо.

4. «Несем» запомненный символ вправо, за знак $=$, за все уже скопированные символы, пока не встретим первую справа пустую ячейку. Вписываем в эту пустую ячейку «принесенный» символ и переходим во внутреннее состояние $\{ \}$. Теперь циклически повторяется все, что описано в конце пункта 1, в пунктах 2, 3 и 4. Символы слева копируются («переносятся») направо. Позиция возврата «за следующим копируемым символом» отмечается символом-двойником.

5. Когда во внутреннем состоянии $[?]$ головка «налетит» на символ $=$, то это будет означать, что все символы слева уже скопированы. Можно завершить работу.

Функциональная схема (программа) машины Тьюринга E может быть такой:

$$\begin{aligned}
q_1a &\rightarrow q_1aR \\
q_1b &\rightarrow q_1bR \\
q_1a_0 &\rightarrow \{J=L \\
\{J}a &\rightarrow \{J}aL \\
\{J}b &\rightarrow \{J}bL \\
\{J}a_0 &\rightarrow \{?\}a_0R \\
\{J}A &\rightarrow \{?\}aR \\
\{J}B &\rightarrow \{?\}bR
\end{aligned}$$

Идем вправо, приписываем в конце знак =, возвращаемся назад, отыскиваем позицию, в которой стоит очередной копируемый символ.

$$\begin{aligned}
\{?\}a &\rightarrow [a\}AR \\
\{?\}b &\rightarrow [b\}BR
\end{aligned}$$

Запоминаем копируемый символ. Отмечаем его позицию на ленте символом-двойником.

$$\begin{aligned}
[a\}a &\rightarrow [a\}aR \\
[a\}b &\rightarrow [a\}bR \\
[a\} = &\rightarrow [a\} =R \\
[a\}a_0 &\rightarrow \{J}aL \\
[b\}a &\rightarrow [b\}aR \\
[b\}b &\rightarrow [b\}bR \\
[b\} = &\rightarrow [b\} =R \\
[b\}a_0 &\rightarrow \{J}bL \\
\{J} = &\rightarrow \{J} =L
\end{aligned}$$

Транспортируем копируемый символ на его место справа, возвращаемся назад, отыскиваем позицию, в которой стоит очередной копируемый символ.

$$\{?\} = \rightarrow q_0 =$$

Когда все символы скопированы, завершаем работу.

Применив машину Тьюринга E к разным словам, можно убедиться, что она корректно перерабатывает их. В случае с пустым входным словом на ленте появляется знак = в качестве выходного слова. Это вполне корректно: пустое слово удвоено, между ним и его копией стоит знак =.