Модификации машин Тьюринга

Реализовать содержательный алгоритм на машине Тьюринга очень трудоёмко, поэтому рассматривают более удобные вариации машины Тьюринга, сводящиеся к ней.

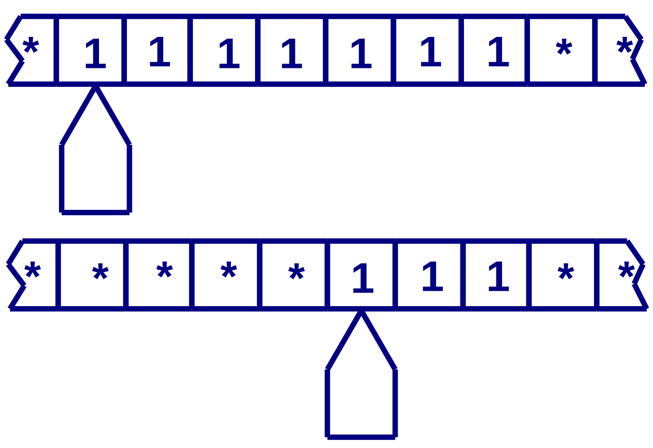
Многоленточные машины Тьюринга

Многоленточная машина для каждой ленты в общем случае может иметь свой внешний алфавит.

Ленты в машине движутся независимо друг от друга.

Состояние для всех лент машины единое, по сути, это состояние управляющего механизма.

Вычислительная способность многоленточных машин совершенно не превосходит их одноленточных аналогов.

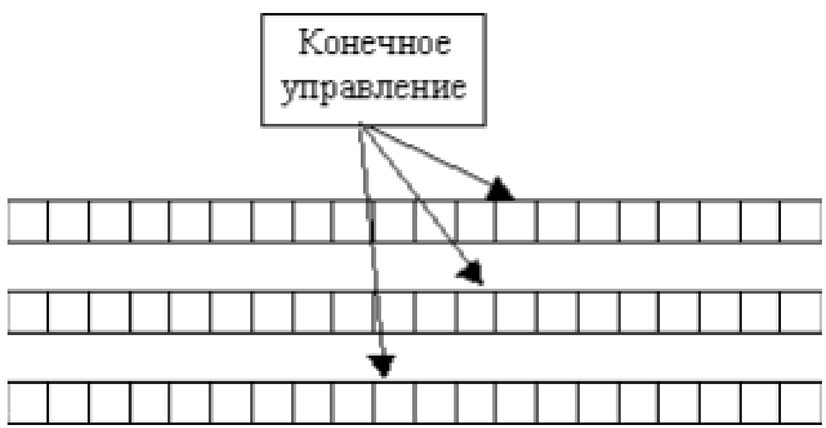


Задачи, которые можно решить на многоленточной машине с произвольным количеством лент, всегда решаются и при помощи одноленточной машины.

Каждая лента бесконечна в обоих направлениях.

При одном движении, зависящем от состояния конечного управления и сканируемого символа каждой из ленточных головок, машина может:

1) изменить состояние;
2) напечатать новый символ на каждой из сканируемых ячеек;
3) передвинуть каждую из ее ленточных головок независимо друг от друга на одну ячейку влево, вправо или оставить ее на том же месте.



Классический формат записи работы многоленточной машины Тьюринга:

$$\mathbf{S_i\{a,b,c\}\rightarrow\{a',b',c'\}\{R,L,H\}S_j.}$$

Машина B ***эквивалентна*** машине A, если в соответствующие такты их работы лента машины B содержит всю информацию о ленте машины A.

Примечание.

Одна из машин может работать гораздо медленнее другой, т.к. каждый такт она моделирует несколькими тактами, поэтому мы говорим о соответствующих тактах. Если в конце концов *A* остановится, то *B* тоже остановится и к этому моменту будет содержать всю информацию о ленте машины *A*.

Теорема. Всякая k - ленточная машина Тьюринга M может преобразована в эквивалентную машину M* с одной лентой.

<u>Доказательство</u>

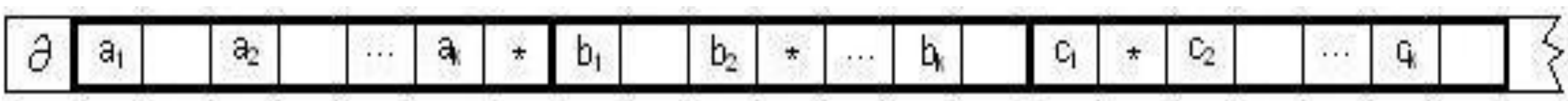
Пусть есть k - ленточная машина M и одноленточная машина M*.

Запишем содержимое лент машины M в виде матрицы с 2k строками и бесконечным числом столбцов. В матрице нечетные строки (1, 3, 5 и т.д. до 2k-1 -ой) занимают непосредственно ленты машины M, а четные (2, 4, 6, ..., 2k-ая) являются служебными. На каждой из служебных строк записан только один символ «*» и его расположение указывает на положение смотрового окошка управляющего механизма на соответствующей ленте.

a_1	b_1	c_1			1-я строка – соотв.1-й ленте машины М
		*			2-я строка – хранит инфо о полож. головки на 1-й ленте м.М
a_2	b_2	c_2			3-я строка – соотв. 2-й ленте машины М
	*				4-я строка – хранит инфо о полож. головки на 2-й ленте м.М
…	…	…			
a_k	b_k	c_k			(2k-1)-я строка – соотв. k-й ленте машины М
*					2k-я строка – хранит инфо о полож. головки на k-й ленте м.М

На 2-ой строке специальный символ * стоит в той клетке, которая находится непосредственно под клеткой, обозреваемой управляющей головкой на первой ленте.

- Построим ленту машины М*. Для этого содержимое матрицы перенесем на ленту одноленточной машины по столбцам: сначала первый столбец, затем второй и т.д. Одна гиперклетка (набор клеток) машины М* равна целому столбцу в таблице



- В этом случае работа машины М* будет повторять работу машины М с тем отличием, что для воспроизведения команды машины М потребуется набор передвижений с целью определения истинной конфигурации. Для этого на ленте машины М* поочередно отыскиваются все символы * и считываются находящиеся слева от них символы внешнего алфавита соответствующих лент.

- Т.о. определяется текущая конфигурация машины М. Далее по программе машины М определяются необходимые действия и они моделируются исходя из формата записи машины М*.
- Например, перемещение управляющего механизма вправо на какой-нибудь из лент имитируется переносом соответствующего символа * на 2k клеток вправо.

- Здесь стоит заметить, что в общем случае, не вполне очевидно, как машина M* будет опознавать на какой из лент находится символ, расположенный левее очередной обнаруженной *. В случае, если алфавиты на всех k лентах различны, это трудностей не составляет и будет учтено при составлении программы соответствующим подбором возможных конфигураций.
- В случае если алфавиты на некоторых или всех лентах совпадают или имеют непустое пересечение, возникает потребность в различных символах, например r_1, r_2, r_3 и т.д. вместо предложенного выше единого символа *.

- Таким образом показано, что k-ленточная машина Тьюринга может быть преобразована в эквивалентную ей одноленточную машину Тьюринга.

Недетерминированная машина Тьюринга

- Недетерминированная машина Тьюринга есть устройство с конечным управлением и одной бесконечной в обе стороны лентой. Для данного состояния и ленточного символа, сканируемого головкой ленты, машина имеет несколько вариантов для следующего движения. Каждый вариант состоит из нового состояния, ленточного символа, который печатается, и направления движения головки.
- Недетерминированная машина Тьюринга принимает входную цепочку, если какая-нибудь последовательность вариантов движений приводит к принимающему состоянию.

При данной входной цепочке x можно считать, что недетерминированная машина Тьюринга M параллельно выполняет все возможные последовательности шагов, пока не достигнет допускающего мгновенного описания или пока не окажется, что дальнейшие шаги невозможны.

- Теорема. Если язык L принимается недетерминированной машиной Тьюринга T_1, то он принимается некоторой детерминированной машиной Тьюринга T_2.

Двумерная машина Тьюринга

- Является еще одной модификацией машины Тьюринга, которая не увеличивает ее мощности.
- Это устройство состоит из обычного конечного управления, но лента разбита на бесконечное число ячеек, расположенных в двух измерениях.
- В зависимости от состояния и сканируемого символа устройство изменяет состояние, печатает новый символ и передвигает ленточную головку в одном из четырех направлений. Первоначально входная цепочка находится на одной строке, а головка находится на левом конце вводной цепочки.
- В любое время только конечное число строк имеет какие-нибудь непустые символы на них, и каждая из этих строк имеет только конечное число непустых символов.

Пример.

Конфигурация двумерной ленты (очерчен прямоугольник вокруг непустых символов):

B	B	B	a_1	B	B	B
B	B	a_2	a_3	a_4	a_5	B
a_6	a_7	a_8	a_9	B	a_{10}	B
B	a_{11}	a_{12}	a_{13}	B	a_{14}	a_{15}
B	B	a_{16}	a_{17}	B	B	B

этот прямоугольник можно записать строка за строкой на одномерной ленте

$$*BBBa_1BBB*BBa_2a_3a_4a_5B*a_6a_7a_8a_9Ba_{10}B*Ba_{11}a_{12}a_{13}Ba_{14}a_{15}*BBa_{16}a_{17}BBB*$$

Символы * разделяют строки. Один из символов помечается как сканируемый головкой. Для пометки можно использовать дополнительную дорожку ленты.

- Если при данном движении головка остается в пределах представленного прямоугольника, то подогнать положение головки нетрудно.
- Если головка выходит за границу прямоугольника при движении в вертикальном направлении, добавляют еще одну строку пробелов к левому или правому концу линейного представления.
- Если головка покидает прямоугольник через правую или левую границу, длина каждой представленной строки должна быть увеличена на единицу. При этом может пригодиться метод “сдвига”.
- Описанный подход легко обобщить на n-мерные ленты.

Машина Тьюринга с входной лентой только для чтения и одной или несколькими лентами памяти для записи/чтения

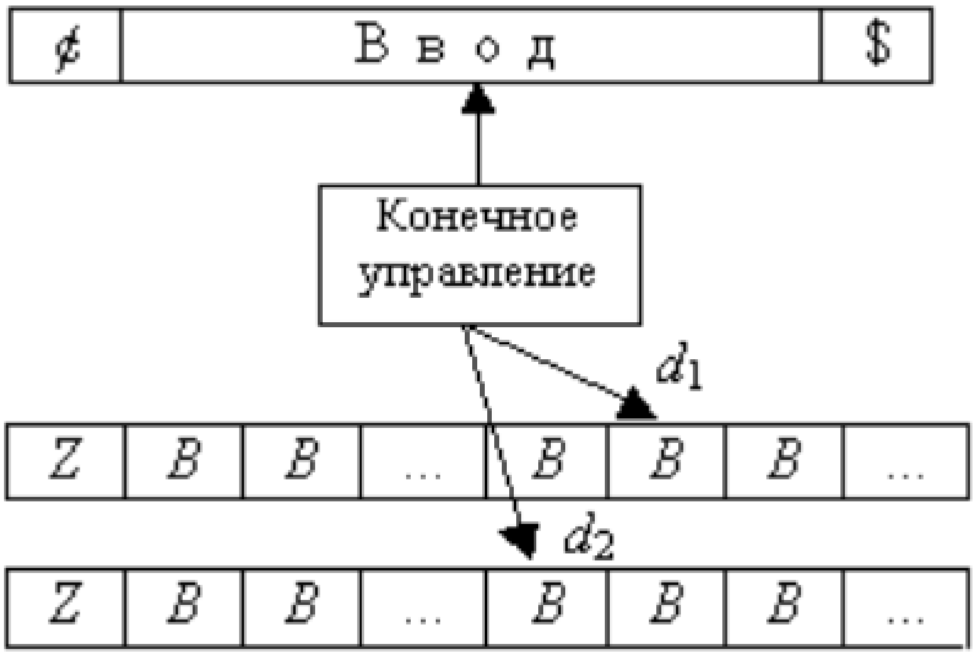
Ее движение зависит от сканируемого входного символа, но она не может печатать на входной ленте. Обычно считается, что входная лента имеет концевые маркеры, так что головка входной ленты может всегда оставаться на входной цепочке, границы которой она не может сама отмечать.

Если входная головка может двигаться в двух направлениях, то устройство называется машиной Тьюринга типа off-line. Если входная головка никогда не движется влево, то это машина Тьюринга типа on-line. Ясно, что машины этих двух типов являются вариантами многоленточных машин Тьюринга. Они могут моделировать любую многоленточную машину Тьюринга.

Ограниченные машины Тьюринга, эквивалентные основной модели

- **Машина со счетчиками** есть машина Тьюринга, алфавиты лент памяти которой содержат только два символа, например Z и B. Кроме того, символ Z появляется в ячейке, первоначально сканируемой ленточной головкой, и никогда не может появиться ни в какой другой. Число i можно запомнить путем передвижения головки ленты на i ячеек вправо от Z. Предполагается, что машины этого типа могут печатать пробел. Запомненное число может быть увеличено или уменьшено передвижением головки ленты вправо или влево.

Пример машины со счетчиком



Символы ¢ и $ обычно используются в качестве концевых маркеров на входной ленте.
Здесь Z — непустой символ.

Конфигурация машины со счетчиками может быть описана при помощи состояния, положения входной головки и расстояний головок лент памяти от символа Z (они обозначены символами d_1 и d_2). Назовем эти расстояния отсчетами на лентах. Машина со счетчиками может фактически только запоминать отсчет на каждой ленте и указывать, является ли этот отсчет нулевым.

Самоанализирующие машины Тьюринга

Знаки таблицы машины Тьюринга:

- *ленточные знаки* – это символы внешнего алфавита,
- *служебные знаки* – это символы внутреннего алфавита (состояния машины), разделитель (он нужен, если таблица записывается в 1 строчку), знаки движения.

Самоанализирующие машины – это машины, в которых все служебные символы как-либо изображаются ленточными символами.

Примечание.

Если в понятие машины Тьюринга включить условие, что при появлении конфигураций, не предусмотренных в таблице машины, она останавливается, то из любой машины Тьюринга легко получить самоанализирующую машину, включив в ее ленточный алфавит все служебные знаки, ранее в нем отсутствующие. Если в первоначальном алфавите их вовсе не было, то новая машина при самоанализе сразу остановится (т.н. тривиальный самоанализ).

Пример

- <u>Допустим</u>:

А – знак единственного внутреннего состояния
R – знак движения вправо
Н – знак остановки
Х – табличный разделитель команд

Αα	Ββ	Γγ	Δδ	Εε	Ζζ
альфа	вита (бета)	гамма	дельта	эпсилон	зита (дзета)
Ηη	Θθ	Ιι	Κκ	Λλ	Μμ
ита (эта)	тита (тета)	йота	каппа	ламбда	ми (мю)
Νν	Ξξ	Οο	Ππ	Ρρ	Σσ
ни (ню)	кси	омикро(н)	пи	ро	сигма
Ττ	Υυ	Φφ	Χχ	Ψψ	Ωω
таф (тау)	ипсило(н)	фи	хи	пси	омега

- Тогда введем ленточные образы этих знаков:

А – α (альфа)
R – ρ (ро)
Н – η (эта)
Х – ξ (кси)

- Можно предложить нарочито упрощенный пример программы машины М.

А α → ξ R A // если машина находится в состоянии А (ленточный образ α), то на это место ставится образ разделителя команд Х (ξ).
А ξ → ξ R A // переход через ленточный образ разделителя команд Х (ξ).
А ρ → η Н А // если машине предписано двигаться вправо, она меняет ленточный образ символа R (ρ) на ленточный образ символа Н (η) и останавливается.

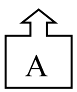
- Посмотрим, как будет проистекать самоанализ машины М.

Образы служебных знаков могут совпадать с самими служебными знаками, но в нашем описании рабочего процесса мы должны их различать.
Код программы машины выглядит так:

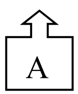
Aα ξRA **X** Aξ ξ RA **X** AρηHA **X**

Тогда на ленте запишем этот же код, заменив знаки ленточными образами (пробелы поставлены для повышения читаемости, на ленте символы реально записаны друг за другом). Машина находится в состоянии A, ее положение на ленте показано графически.

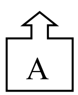
1 шаг: α α ξ ρ α ξ α ξ ξ ρ α ξ α ρ η η α ξ



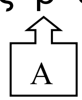
2 шаг: ξ α ξ ρ α ξ α ξ ξ ρ α ξ α ρ η η α ξ



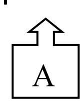
3 шаг: ξ ξ ξ ρ α ξ α ξ ξ ρ α ξ α ρ η η α ξ



4 шаг: ξ ξ ξ ρ α ξ α ξ ξ ρ α ξ α ρ η η α ξ



5 шаг: ξ ξ ξ η α ξ α ξ ξ ρ α ξ α ρ η η α ξ



Универсальные машины

Универсальная машина (УМТ) – машина Тьюринга, обладающая способностью путём подходящего кодирования выполнить любое вычисление.

Пусть машина A имеет m символов a_j и n внутренних состояний S_i. Закодируем знаки, используемые при написании программы работы такой машины следующим образом.

$a_j = 1\ldots1$ ($a_1=1$, $a_2=11$, $a_3=111$ и т.д.)

$S_i = 2\ldots2$ ($S_1=2$, $S_2=22$, $S_3=222$ и т.д.)

$R = 3$

$L = 33$

$H = 333$

В этом случае всю программу работы машины можно записать неким числом, причем возможны два варианта записи:

1. С разделителем команд, допустим X, которые можно закодировать числом 4. В этом случае классическая запись $S_{old}\ a_{old} \rightarrow a_{new}\ R\ S_{new}$,
2. Без разделителя команд. В этом случае команды следует писать в формате $a_{old}\ S_{old}\ a_{new}\ R\ S_{new}$, тогда две команды, записанные непосредственно друг за другом, будут явно различаться элементарным анализатором.

Пример: Машина Тьюринга, которая на пустой ленте бесконечно много раз печатает последовательность *001*.

(Из соображений удобства формат записи $aS_i \rightarrow a'\{R,L,H\}S_j$)

- Программа такой машины выглядит следующим образом (λ-пустой знак):

λ A 0 R B

λ B 0 R C

λ C 1 R A

- Закодируем символы и состояния:

λ=1 0=11 1=111   A=2   B=22   C=222

- Тогда запись программы машины будет выглядеть так (пробелы поставлены для повышения читаемости, в реальности их нет):

1 2 11 3 22   1 22 11 3 222   1 222 111 3 2

- Т.о. каждая машина Тьюринга представлена числом – это дескриптивное (описательное) число машины. Вместе с тем оно является кодом для входного слова.

Пример построения универсальной машины Тьюринга

- Рассмотрим УМТ. Тогда, если цифру 5 использовать для разделения описания машины и описания входного слова, лента выглядит следующим образом

∂	dT	5	dW				

dT – описание машины (алфавит 1, 2, 3)

dW – описание входного слова, в котором каждый символ слова a_i записан в виде наборов по несколько *1*, разделенных специальным символом *4* (разделитель).

- Описание машины – слово, разбитое на команды. УМТ читает описание данной машины, а затем перерабатывает входное слово так, как бы это сделала конкретная МТ. УМТ имеет такой же алфавит, как и у предъявленной ей машины.
- Отсюда вытекает необходимость наличия меток для указания положения исходной МТ. Такие метки удобно ставить вместо разделителя, стоящего непосредственно перед группой ячеек, содержащих код рассматриваемого символа. Поскольку цифры {1,2,3,4,5} алфавита уже заняты, будем заменять разделитель *4* перед символом, на котором остановилась исходная МТ, на цифру *6*.

Пример построения универсальной машины Тьюринга

- Составление таблицы для одноленточной УМТ длительная и малопригодная для понимания в учебных целях процедура. Поэтому для наглядности построим трехленточную машину, которую всегда можно преобразовать в одноленточную по принципу, рассмотренному в предыдущей теореме.

- Ленты трехленточной универсальной машины Тьюринга будут иметь следующее назначение:
  - исходная лента, на которой записан код таблицы исходной МТ,
  - рабочая лента, на ней записываются внутренние состояния исходной МТ,
  - выходная лента – закодированная лента исходной МТ.

- В начале моделирования каждого такта работы исходной МТ, УМТ занимает на 3-ей ленте первую клетку.
- Эта клетка соответствует той клетке на ленте исходной МТ, которую в данный момент воспринимает считывающая головка исходной МТ.

Пример построения универсальной машины Тьюринга

- УМТ движется по входной ленте, пока не дойдёт до команды, в которой внутреннее состояние совпадает с записью на 2-ой ленте, а воспринимаемый символ – с тем, который записан на входной ленте, в кодовой ячейке которой изображено положение исходной машины.

- Сравнение происходит по разрядам. В итоге УМТ находит на входной ленте нужную для исполнения команду МТ.

- Очень важен механизм опознания того факта, что внутреннее состояние совпадает с записью на 2-ой ленте, а воспринимаемый символ – с тем, который записан на входной ленте. Любое запоминание прочитанных символов может производиться только путем изменения внутреннего состояния, однако при программировании УМТ количество состояний должно быть ограничено. В этой связи метод введения все новых и новых состояний типа S^{1} S^{11} S^{111} S^{111}_{2} S^{111}_{22} $S^{111}_{222}S^{111}_{2222}$ и т.д. для запоминания количества «2» на второй ленте и «1» на третьей ленте является неосуществимым. Нам просто не хватит описательных возможностей для перебора всех допустимых состояний. Реально требуется механизм ***поразрядного*** сравнения. Способов естественно огромное множество.

Пример построения универсальной машины Тьюринга

- По сути, на первой ленте нужно найти некий набор последовательностей «1» и «2», идущих сразу после разделителя команд, который соответствует находящемуся на второй ленте коду состояния исходной машины (последовательность «2») и обозреваемому на третьей ленте коду символа (последовательность «1»). Для оптимизации поэтапного сравнения возможно первоначально стоит проверять совпадение символа (т.е. двигать смотровые окошки на 1-ой и 3-ой ленте) и в случае удачи, продолжить сравнение, двигая окошки на 1-ой и 2-ой ленте.

- Вероятны и другие способы поразрядного сравнения (например, копирование текущей конфигурации на 4-ую ленту и поиск соответствия её содержимому на 1-ой ленте), но важным остается сам факт: запоминать слово неопределенной длины целиком невозможно, в рамках наших возможностей только поразрядное сравнение содержимого лент.
- Так или иначе, УМТ считывает инструкцию, соответствующую этой команде. Затем эта инструкция выполняется,
- при этом совершается **три** действия.

Пример построения универсальной машины Тьюринга

1

- Изменяется положение исходной МТ. Например, если инструкция гласит «шаг вправо», то на 3-ей ленте делается необходимое количество шагов вправо до ближайшего нового разделителя команд, который заменяется меткой текущего положения (вместо 4 ставится символ 6).

2

- Изменяется внутреннее состояние исходной МТ. На второй ленте вместо старого записывается ее новое состояние.

3

- Изменяется символ, содержащийся в рассматриваемой ячейке. Поскольку разные символы кодируются различным количеством единичек, то в случае замены символа необходимо предусмотреть процедуру сдвига (растягивание или сжимание слова на нужное число клеток).

После этого моделируется следующий такт работы исходной МТ. Для этого снова анализируется содержание второй ленты (это теперь текущее состояние МТ) и символ, записанный справа от указателя текущего положения управляющей головки (специальная метка 6).

Универсальная машина М.Минского

	1	2	3	4	5	6	7
y	0L1	0L1	yL3	yL4	yR5	yR6	0R7
0	0L1	yR2	0S0	yR5	yL3	AL3	yR6
1	1L2	AR2	AL3	1L7	AR5	AR6	1R7
A	1L1	yR6	1L4	1L4	1R5	1R6	0R2

Машина – однолеточная.
В ней семь состояний и четыре символа

ВСЕГО ЛИШЬ 28 строк!

Источник: Минский М. Вычисления и автоматы.
М: Мир, 1971, с.331

Тезис Тьюринга

Всякий алгоритм представим в виде машины Тьюринга.

Любая функция, которая может быть вычислена физическим устройством, может быть вычислена машиной Тьюринга.

Замечание:
Также как и тезис Чёрча, тезис Тьюринга недоказуем и является принимаемым без доказательства фундаментальным положением теории алгоритмов.

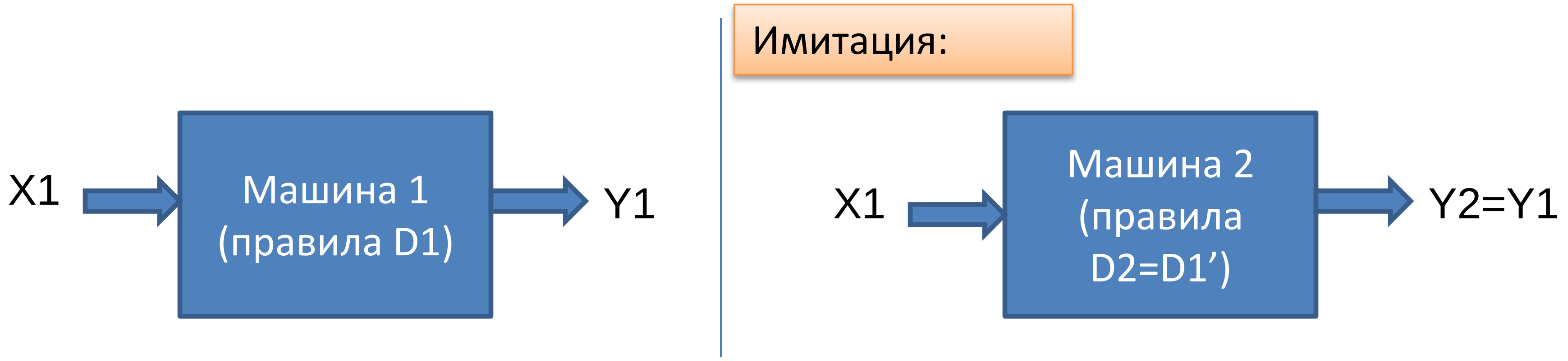
Замечание:
Уверенность в истинности тезиса Тьюринга основана на опыте: пока не найден алгоритм, который не может быть записан в виде МТ.

Тезис Тьюринга (полнота по Тьюрингу)

Один из естественных способов доказательства того, что алгоритмы вычисления, которые можно реализовать на одной машине, можно реализовать и на другой, — это имитация первой машины на второй.

Имитация:

На вход второй машине подаётся описание программы (правил работы) первой машины *D1* и входные данные *X1*, которые должны были поступить на вход первой машины. Нужно описать такую программу (правила работы второй машины), чтобы в результате вычислений на выходе оказалось Y2 то же самое, что вернула бы первая машина Y1 (Y1=Y2), если бы получила на вход данные *X1*.



Тезис Тьюринга (полнота по Тьюрингу)

На машине Тьюринга можно имитировать:

машину Поста,
нормальные алгоритмы Маркова,
машины с арх. фон-Неймана
и др.

В свою очередь, на различных абстрактных исполнителях можно имитировать Машину Тьюринга. Исполнители, для которых это возможно, называются *полными по Тьюрингу*