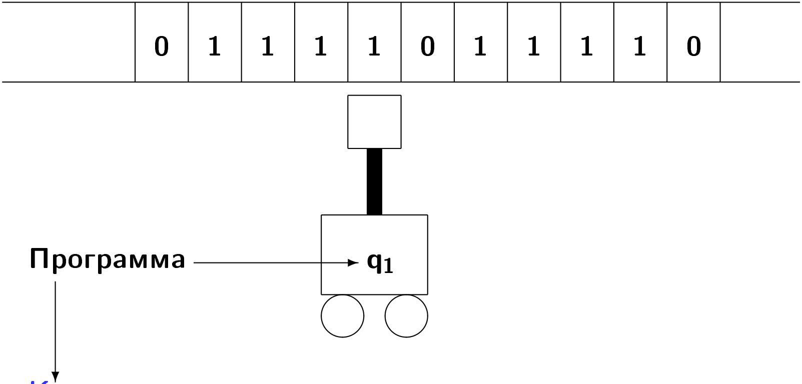
# Модели вычислений

В.А. Захаров, Р.И. Подловченко

# Лекция 4.

1. Машины Тьюринга.
2. Рекурсивные и рекурсивно перечислимые языки
3. Вариации машин Тьюринга
4. Свойства замкнутости
5. Функции, вычислимые по Тьюрингу
6. Массовые алгоритмические проблемы

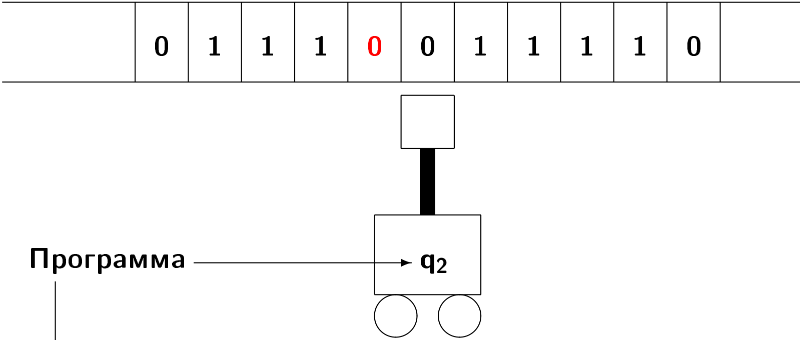
# МАШИНЫ ТЬЮРИНГА




Команды:

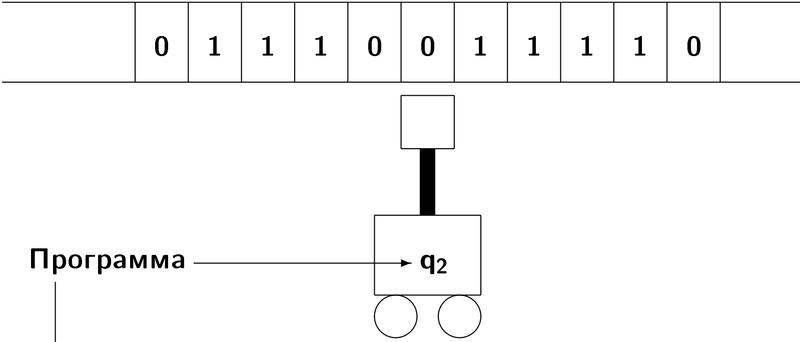
увидев "1" , стереть его, записать "0" и сдвинуться вправо

# МАШИНЫ ТЬЮРИНГА




Команды:

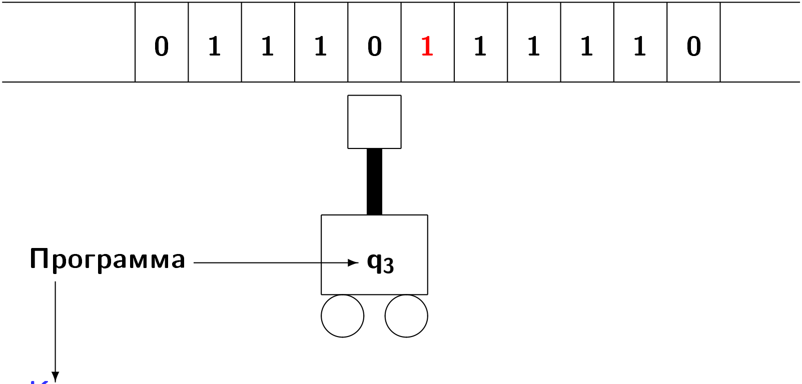
# МАШИНЫ ТЬЮРИНГА




**Команды:**

увидев "0" , стереть его, записать "1" и сдвинуться влево

# Машины Тьюринга




Команды:

# МАШИНЫ ТЬЮРИНГА

Пусть заданы конечные входной алфавит $\Sigma = \{a_1, a_2, \ldots, a_n\}$ и ленточный алфавит $\Gamma$, $\Sigma \subseteq \Gamma$, в котором особо выделен пустой символ $0$, $0 \notin \Sigma$.

Входным словом (ленточным словом) будем называть всякое слово в алфавите $\Sigma$ (соответственно $\Gamma$).

Пусть задан конечный алфавит состояний $Q = \{q_{-1}, q_0, q_1, q_2, \ldots, q_m\}$, $Q \cap \Gamma = \varnothing$, в котором особо выделены начальное состояние $q_1$, допускающее состояние $q_0$, отвергающее состояние $q_{-1}$.

# МАШИНЫ ТЬЮРИНГА

Машиной Тьюринга (МТ) называется система $\mathcal{M} = (\Sigma, \Gamma, Q, q_1, q_0, q_{-1}, T)$, где

$$T : Q \setminus \{q_0, q_{-1}\} \times \Gamma \;\to\; \Gamma \times Q \times \{L, R\}$$

всюду определенная функция переходов.

Функцию переходов МТ удобно представлять в виде множества наборов

$$\{(q, x : y, q', D) : T(q, x) = (y, q', D), q \in Q, x \in \Gamma\},$$

которые будем называть командами МТ.

# Машины Тьюринга

Каждую команду $(q, x : y, q', D)$ нужно понимать так:

если МТ находится в состянии $q$ и обозревает символ $x$, то записать в обозреваемую ячейку символ $y$, перейти в состояние $q'$ и сдвинуть считывающую головку на одну ячейку в направлении $D$.

# МАШИНЫ ТЬЮРИНГА

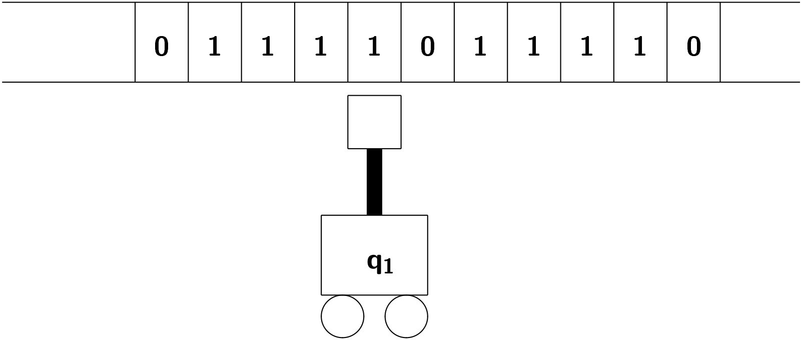
МТ работает на бесконечной ленте, в которой почти все ячейки заполнены пустым символом $0$. Вычисления МТ можно описать в терминах ленточных конфигураций.

Ленточной конфигурацией МТ $\mathcal{M}$ называется всякое слово $(w'qxw'')$, где $w', w'' \in \Gamma^*$, $q \in Q$, $x \in \Gamma$. Такая конфигурация содержательно означает, что

- $q$ — это то состояние, в котором находится МТ,
- $x$ — это буква, которая записана в той ячейке ленты, которую обозревает считывающая головка МТ,
- $w'$ — это ленточное слово, составленное из символов, записанных **слева** от обозреваемой ячейки,
- $w''$ — это ленточное слово, составленное из символов, записанных **справа** от обозреваемой ячейки.

По умолчанию считается, что во всех остальных ячейках ленты записаны пустые символы.

# МАШИНЫ ТЬЮРИНГА




Ленточная конфигурация: $0111q_1 1011110$

# Машины Тьюринга

Ленточная конфигурация вида $u\ q_1\ x\ v$ называется начальной конфигурацией.

Ленточная конфигурация вида $u\ q_0\ x\ v$ называется допускающей конфигурацией.

Ленточная конфигурация вида $u\ q_{-1}\ x\ v$ называется отвергающей конфигурацией.

# МАШИНЫ ТЬЮРИНГА

Каждая команда $K$ задает отношение перехода $\xrightarrow{K}$ на множестве ленточных конфигураций:

$$\alpha \xrightarrow{K} \beta$$

Оно определяется так:

если $\alpha = uzqxv$ и $K = qx : yq'L$, то $\beta = uq'zyv$,

если $\alpha = qxv$ и $K = qx : yq'L$, то $\beta = q'0yv$,

если $\alpha = uqxzv$ и $K = qx : yq'R$, то $\beta = uyq'zv$,

если $\alpha = uqx$ и $K = qx : yq'R$, то $\beta = uyq'0$.

# МАШИНЫ ТЬЮРИНГА

Каждая команда $K$ задает отношение перехода $\xrightarrow{K}$ на множестве ленточных конфигураций:

$$\alpha \xrightarrow{K} \beta$$

Оно определяется так:

если $\alpha = uzqxv$ и $K = qx : yq'L$, то $\beta = uq'zyv$,
если $\alpha = qxv$ и $K = qx : yq'L$, то $\beta = q'0yv$,
если $\alpha = uqxzv$ и $K = qx : yq'R$, то $\beta = uyq'zv$,
если $\alpha = uqx$ и $K = qx : yq'R$, то $\beta = uyq'0$.

Программа $\mathcal{M}$ задает отношение переходов $\xrightarrow{\mathcal{M}}$ на множестве ленточных конфигураций:

$$\xrightarrow{\mathcal{M}} = \bigcup_{K \in T} \xrightarrow{K}.$$

# МАШИНЫ ТЬЮРИНГА

Вычислением МТ $\mathcal{M}$ из начальной конфигурации $\alpha_0$ называется последовательность конфигураций

$$\mathcal{M}(\alpha_0) = \alpha_0 \xrightarrow{\mathcal{M}} \alpha_1 \xrightarrow{\mathcal{M}} \alpha_2 \xrightarrow{\mathcal{M}} \cdots$$

которая либо является бесконечной, либо завершается допускающей или отвергающей конфигурацией.

Вычисление, оканчивающееся допускающей конфигурацией, называется допускающим .

Вычисление, оканчивающееся отвергающей конфигурацией, называется отвергающим .

# РЕКУРСИВНЫЕ и РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ ЯЗЫКИ

Язык МТ $\mathcal{M}$ — это множество $L(\mathcal{M})$ всех слов $w, w \in \Sigma^*$, для которых вычисление из начальной конфигурации $\alpha_0 = q_1 w 0$ является допускающим:

$$\alpha_0 = q_1 w 0 \xrightarrow{\mathcal{M}}_* \alpha_N = u q_0 v.$$

# РЕКУРСИВНЫЕ и РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ ЯЗЫКИ

Язык МТ $\mathcal{M}$ — это множество $L(\mathcal{M})$ всех слов $w, w \in \Sigma^*$, для которых вычисление из начальной конфигурации $\alpha_0 = q_1 w 0$ является допускающим:

$$\alpha_0 = q_1 w 0 \xrightarrow{\mathcal{M}}_* \alpha_N = u q_0 v.$$

Язык $L$ называется рекурсивно перечислимым, если $L = L(\mathcal{M})$ для некоторой МТ $\mathcal{M}$.

Язык $L$ называется рекурсивным, если $L = L(\mathcal{M})$ для некоторой МТ $\mathcal{M}$, которая обладает свойством тотальности, т.е. для любого входного слова $w, w \in \Sigma^*$, вычисление $\mathcal{M}$ из начальной конфигурации $\alpha_0 = q_1 w 0$ является конечным.

# Рекурсивные и рекурсивно перечислимые языки

Рекурсивность языка $L$ означает, что есть такая МТ $\mathcal{M}$, которая для каждого входного слова $w, w \in \Sigma^*$, может спустя конечное число шагов вычисления правильно ответить на вопрос, принадлежит ли $w$ языку $L$ либо утвердительно (вычисление $\mathcal{M}(q_0 w 0)$ — допускающее), либо отрицательно (вычисление $\mathcal{M}(q_0 w 0)$ — отвергающее), т.е. дать алгоритмическое решение задачи проверки принадлежности слова языку.

Потому рекурсивные языки также называют (алгоритмически) разрешимыми .

# РЕКУРСИВНЫЕ И РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ ЯЗЫКИ

**Примеры** рекурсивных языков

1. $L = \varnothing$ и $L = \Sigma^*$;

# Рекурсивные и рекурсивно перечислимые языки

**Примеры** рекурсивных языков

1. $L = \varnothing$ и $L = \Sigma^*$;
2. любой регулярный язык $L$;

# Рекурсивные и рекурсивно перечислимые языки

**Примеры** рекурсивных языков

1. $L = \varnothing$ и $L = \Sigma^*$;
2. любой регулярный язык $L$;
3. $L = \{ww : w \in \Sigma^*\}$;

# Рекурсивные и рекурсивно перечислимые языки

**Примеры** рекурсивных языков

1. $L = \varnothing$ и $L = \Sigma^*$;
2. любой регулярный язык $L$;
3. $L = \{ww : w \in \Sigma^*\}$;
4. $L = \{a^p : p \text{ — простое число}\}$;

# РЕКУРСИВНЫЕ И РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ ЯЗЫКИ

**Примеры** рекурсивных языков

1. $L = \varnothing$ и $L = \Sigma^*$;
2. любой регулярный язык $L$;
3. $L = \{ww : w \in \Sigma^*\}$;
4. $L = \{a^p : p$ — простое число$\}$;
5. $L = \{w : w$ — синтаксически корректный текст С-программы$\}$

# РЕКУРСИВНЫЕ И РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ ЯЗЫКИ

Рекурсивная перечислимость языка $L$ означает, что есть такая МТ $\mathcal{M}$, которая для каждого входного слова $w, w \in \Sigma^*$, спустя конечное число шагов вычисления дает утвердительный ответ на вопрос, принадлежит ли $w$ языку $L$ в том и только том случае, если $w \in L$.

Однако для тех слов, которые не принадлежат языку $L$, МТ $\mathcal{M}$ не обязана давать отрицательный ответ; в этих случаях она может иметь бесконечное вычисление (зацикливается).

# Рекурсивные и рекурсивно перечислимые языки

Поскольку в случае $L = L(\mathcal{M})$ МТ $\mathcal{M}$ гарантирует лишь частичное (только утвердительное) решение задачи о принадлежности слова языку, рекурсивно перечислимые языки также называются полуразрешимыми языками.

# Рекурсивные и рекурсивно перечислимые языки

Поскольку в случае $L = L(\mathcal{M})$ МТ $\mathcal{M}$ гарантирует лишь частичное (только утвердительное) решение задачи о принадлежности слова языку, рекурсивно перечислимые языки также называются полуразрешимыми языками.

Из определения рекурсивных и рекурсивно перечислимых языков ясно видно, что каждый рекурсивный язык является рекурсивно перечислимым.

# Рекурсивные и рекурсивно перечислимые языки

Но есть также важные вопросы, ответы на которых неочевидны.

# РЕКУРСИВНЫЕ И РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ ЯЗЫКИ

Но есть также важные вопросы, ответы на которых неочевидны.

**Существуют ли языки, не являющиеся рекурсивно перечислимыми?**

**Существуют ли рекурсивно перечислимые языки, не являющиеся рекурсивными?**

# Рекурсивные и рекурсивно перечислимые языки

На первый вопрос можно дать утвердительный ответ, не привлекая глубоких исследований.

**Теорема 4.1.** Существуют языки, не являющиеся рекурсивно перечислимыми.

# Рекурсивные и рекурсивно перечислимые языки

На первый вопрос можно дать утвердительный ответ, не привлекая глубоких исследований.

**Теорема 4.1.** Существуют языки, не являющиеся рекурсивно перечислимыми.

**Доказательство.** 1). Каждый рекурсивно перечислимый язык $L$ состоит из всех слов, допускаемых некоторой МТ $\mathcal{M}$.

Каждая МТ $\mathcal{M}$ полностью описывается набором команд $T_{\mathcal{M}}$ этой машины, и этот набор команд представляет собой слово некоторого фиксированного конечного алфавита.

Значит, каждый рекурсивно перечислимый язык описывается некоторым словом конечного алфавита.

Из теории множеств известно, что множество всех слов конечного алфавита счетно. Таким образом, существует только счетное множество рекурсивно перечислимых языков.

# РЕКУРСИВНЫЕ И РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ ЯЗЫКИ

**Доказательство.** 2). Язык — это произвольное подмножество множества входных слов $\Sigma^*$ .

Из теории множеств известно, что множество всех слов $\Sigma^*$ счетно. Из теории множеств также известно, что семейство всех подмножеств счетного множества, не является счетным. Таким образом, множество всех языков несчетно.

# Рекурсивные и рекурсивно перечислимые языки

**Доказательство.** 2). Язык — это произвольное подмножество множества входных слов $\Sigma^*$.

Из теории множеств известно, что множество всех слов $\Sigma^*$ счетно. Из теории множеств также известно, что семейство всех подмножеств счетного множества, не является счетным. Таким образом, множество всех языков несчетно.

Значит, из сравнения мощности семейства всех языков и семейства рекурсивно перечислимых языков следует, что существуют языки, не являющиеся рекурсивно перечислимыми.

QED

# РЕКУРСИВНЫЕ И РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ ЯЗЫКИ

А вот для ответа на второй вопрос

**Существуют ли рекурсивно перечислимые языки, не являющиеся рекурсивными?**

математикам пришлось создать целую теорию, которая существенно повлияла на развитие нашей цивилизации.

# Рекурсивные и рекурсивно перечислимые языки

А вот для ответа на второй вопрос

**Существуют ли рекурсивно перечислимые языки, не являющиеся рекурсивными?**

математикам пришлось создать целую теорию, которая существенно повлияла на развитие нашей цивилизации.

Но вначале попробуем выяснить, насколько существенно изменяются вычислительные возможности машин Тьюринга при внесении некоторых изменений в их устройство.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

Попробуем вначале ослабить вычислительные возможности машин Тьюринга, упрощая их устройство.

Рассмотрим вычислительное устройство, отличающееся от «обычной» машины Тьюринга лишь тем, что лента, на которой работает это устройство, бесконечна лишь в одну сторону (например, вправо). Назовем это устройство односторонней машиной Тьюринга .

Для односторонних МТ можно также определить язык, состоящий из входных слов, на которых МТ завершает вычисление в допускающем состоянии.

**Теорема 4.2.**
1). Для любой 1-сторонней МТ $\mathcal{M}_1$ существует такая 2-сторонняя МТ $\mathcal{M}_2$, что $L(\mathcal{M}_1) = L(\mathcal{M}_2)$.

2). Для любой 2-сторонней МТ $\mathcal{M}_2$ существует такая 1-сторонняя МТ $\mathcal{M}_1$, что $L(\mathcal{M}_2) = L(\mathcal{M}_1)$.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

**Теорема 4.2.**
1). Для любой 1-сторонней МТ $\mathcal{M}_1$ существует такая 2-сторонняя МТ $\mathcal{M}_2$, что $L(\mathcal{M}_1) = L(\mathcal{M}_2)$.

2). Для любой 2-сторонней МТ $\mathcal{M}_2$ существует такая 1-сторонняя МТ $\mathcal{M}_1$, что $L(\mathcal{M}_2) = L(\mathcal{M}_1)$.

**Доказательство.** 1). Очевидно.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

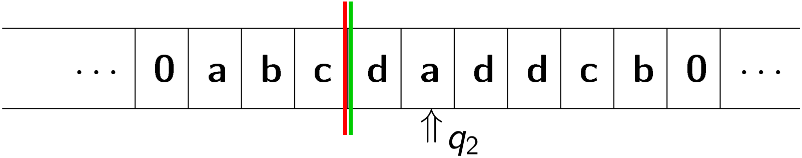
**Доказательство.** 2). Рассмотрим 2-стороннюю МТ $\mathcal{M}_2$ .

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\cdots$ | **0** | **a** | **b** | **c** | **d** | **a** | **d** | **d** | **c** | **b** | **0** | $\cdots$ |

$\Uparrow q_2$

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

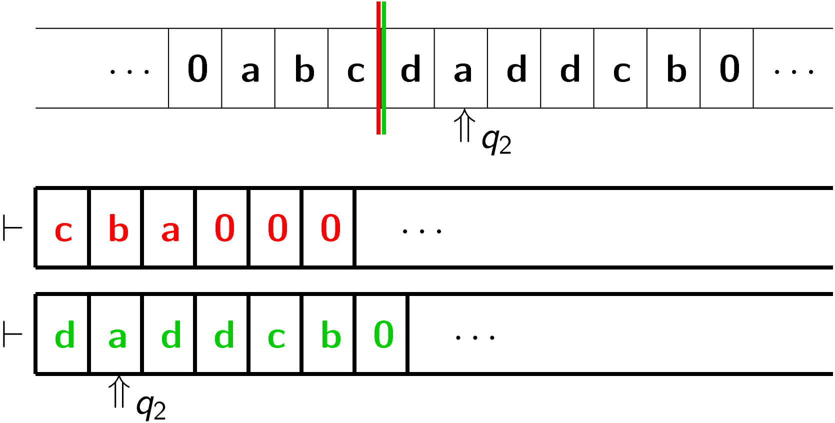
**Доказательство.** 2). Рассмотрим 2-стороннюю МТ $\mathcal{M}_2$ .




Разделим ленту на две половины...

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

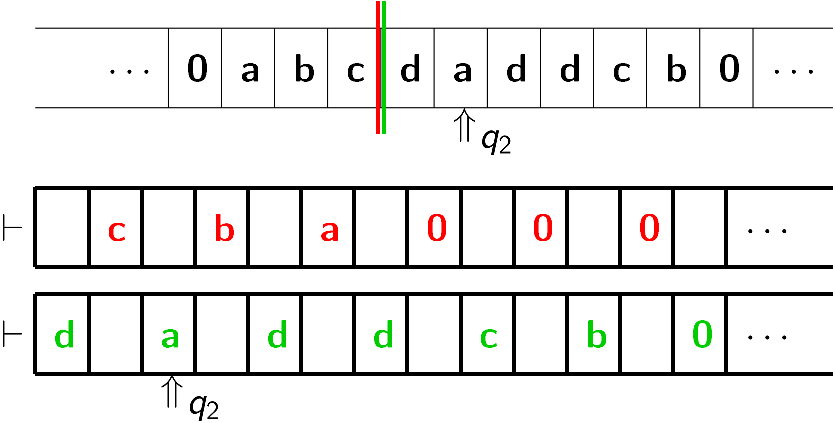
**Доказательство.** 2). Рассмотрим 2-стороннюю МТ $\mathcal{M}_2$ .




Разделим ленту на две половины
и разорвем ее пополам.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

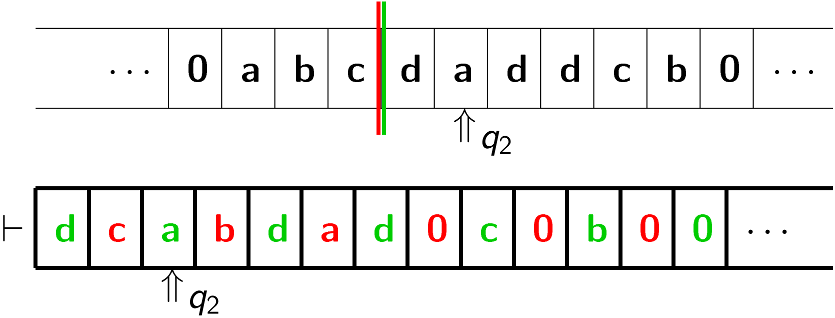
**Доказательство.** 2). Рассмотрим 2-стороннюю МТ $\mathcal{M}_2$ .




Потом раздвинем содержимое обеих лент и...

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

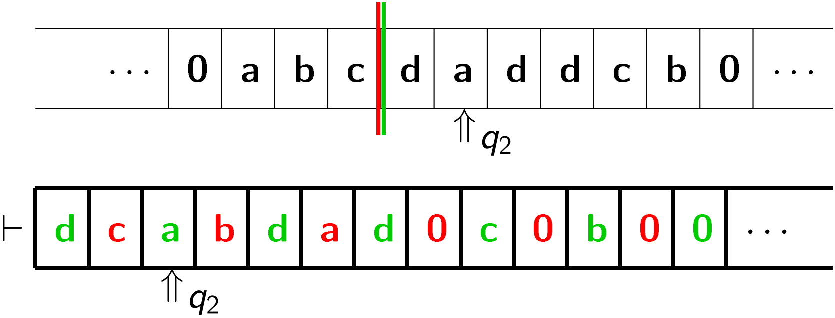
**Доказательство.** 2). Рассмотрим 2-стороннюю МТ $\mathcal{M}_2$ .




Потом раздвинем содержимое обеих лент и разместим его на одной ленте.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

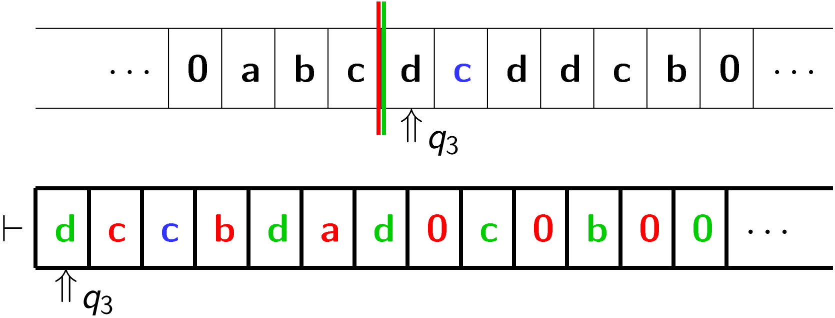
**Доказательство.** 2). Рассмотрим 2-стороннюю МТ $\mathcal{M}_2$ .




Получим одноленточную МТ $\mathcal{M}_1$ , которая может симулировать вычисления МТ $\mathcal{M}_2$ .

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

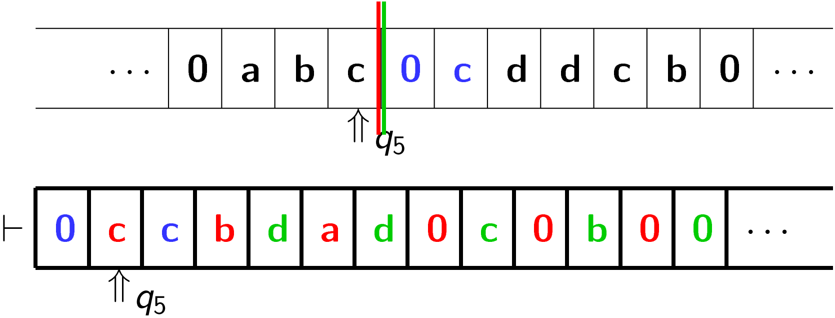
**Доказательство.** 2). Рассмотрим 2-стороннюю МТ $\mathcal{M}_2$ .




Получим одноленточную МТ $\mathcal{M}_1$ , которая может симулировать вычисления МТ $\mathcal{M}_2$ .

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

**Доказательство.** 2). Рассмотрим 2-стороннюю МТ $\mathcal{M}_2$ .




Получим одноленточную МТ $\mathcal{M}_1$ , которая может симулировать вычисления МТ $\mathcal{M}_2$ .

# Вариации машин Тьюринга

Таким образом, односторонние машины Тьюринга обладают точно такими же вычислительными возможностями, как и двусторонние («обычные») машины Тьюринга.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

Таким образом, односторонние машины Тьюринга обладают точно такими же вычислительными возможностями, как и двусторонние («обычные») машины Тьюринга.

А теперь попробуем усилить вычислительные возможности машин Тьюринга.

Рассмотрим вычислительное устройство, отличающееся от «обычной» машины Тьюринга лишь тем, что оно работает на нескольких лентах. На каждой ленте есть своя считывающая головка. Каждое состояние $q$ такой МТ приписано к одной из лент, и в этом состоянии МТ управляет считывающей головкой соответствующей ленты.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

Машины Тьюринга такого вида называются многоленточными МТ.

Одна из лент многоленточной машины выделена как входная лента. В начале вычисления на входной ленте записывается входное слово. Остальные ленты называются рабочими ; в начале вычисления рабочие ленты заполнены пустыми буквами.

Для многоленточных МТ можно также определить язык, состоящий из входных слов, на которых МТ завершает вычисление в допускающем состоянии.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

**Теорема 4.3.**

1). Для любой 1-ленточной МТ $\mathcal{M}_1$ существует такая n-ленточная МТ $\mathcal{M}_n$, что $L(\mathcal{M}_1) = L(\mathcal{M}_n)$.

2). Для любой n-ленточная МТ $\mathcal{M}_n$ существует такая 1-ленточная МТ $\mathcal{M}_1$, что $L(\mathcal{M}_n) = L(\mathcal{M}_1)$.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

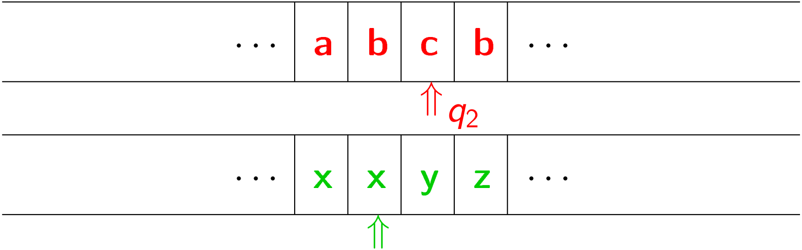
**Теорема 4.3.**
1). Для любой 1-ленточной МТ $\mathcal{M}_1$ существует такая n-ленточная МТ $\mathcal{M}_n$, что $L(\mathcal{M}_1) = L(\mathcal{M}_n)$ .

2). Для любой n-ленточная МТ $\mathcal{M}_n$ существует такая 1-ленточная МТ $\mathcal{M}_1$, что $L(\mathcal{M}_n) = L(\mathcal{M}_1)$ .

**Доказательство.** 1). Очевидно.

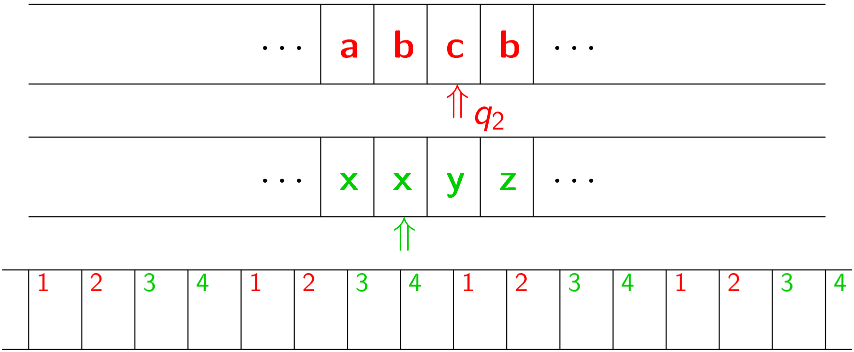
# ВАРИАЦИИ МАШИН ТЬЮРИНГА

2) Рассмотрим 2-ленточную МТ $\mathcal{M}_2$ .



# ВАРИАЦИИ МАШИН ТЬЮРИНГА

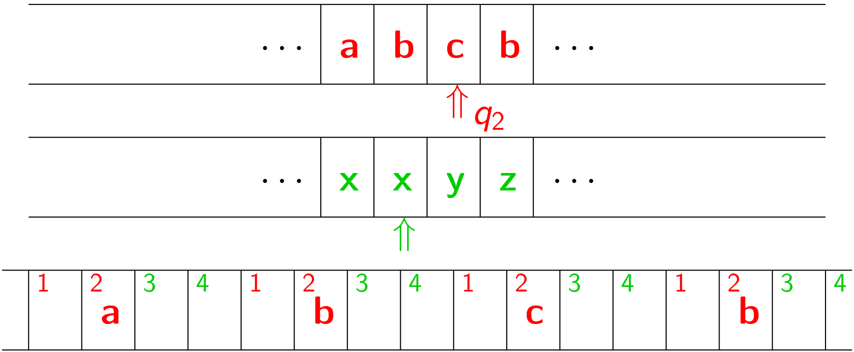
2) Рассмотрим 2-ленточную МТ $\mathcal{M}_2$ .




Чтобы построить 1-ленточную МТ,
разобьем ее ленту на 4 решетки,

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

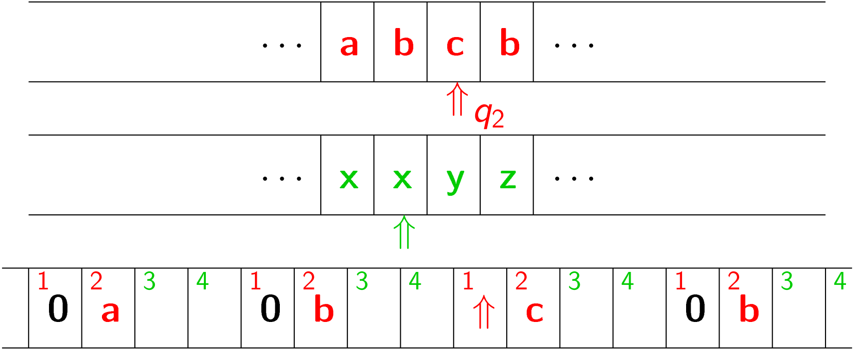
2) Рассмотрим 2-ленточную МТ $\mathcal{M}_2$ .




в ячейках решетки 2 поместим
содержимое 1-ой ленты МТ $\mathcal{M}_2$ ,

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

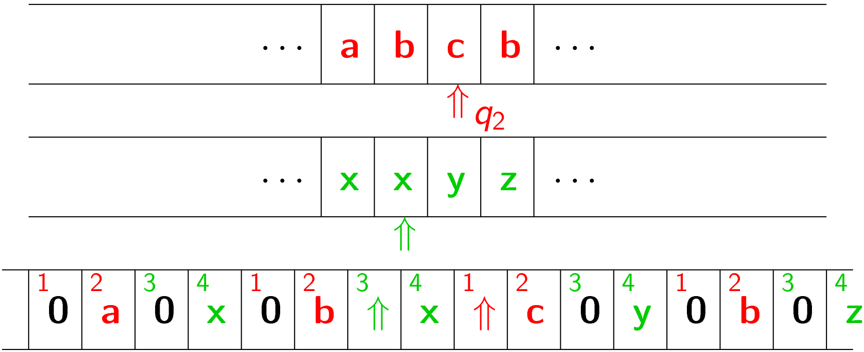
2) Рассмотрим 2-ленточную МТ $\mathcal{M}_2$ .




а в одной из ячеек решетки 1 отметим положение считывающей головки на 1-ой ленты МТ $\mathcal{M}_2$ .

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

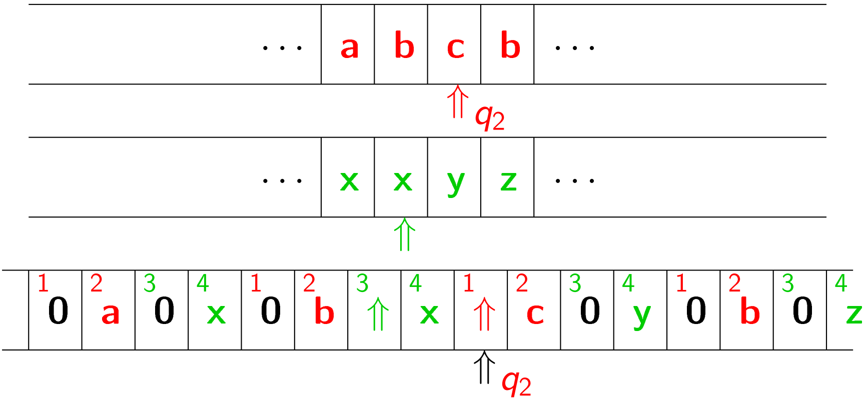
2) Рассмотрим 2-ленточную МТ $\mathcal{M}_2$ .




То же самое проделаем на решетках 3 и 4
для содержимого 2-ой ленты МТ $\mathcal{M}_2$

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

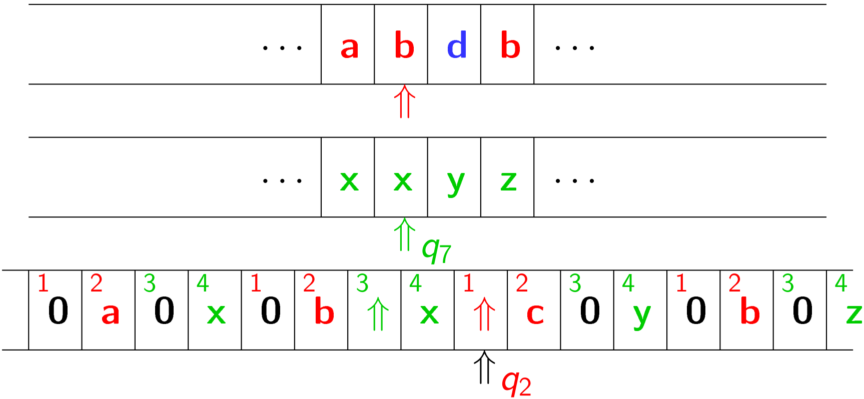
2) Рассмотрим 2-ленточную МТ $\mathcal{M}_2$ .




Получим одноленточную МТ $\mathcal{M}_1$ , которая может симулировать вычисления МТ $\mathcal{M}_2$ .

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

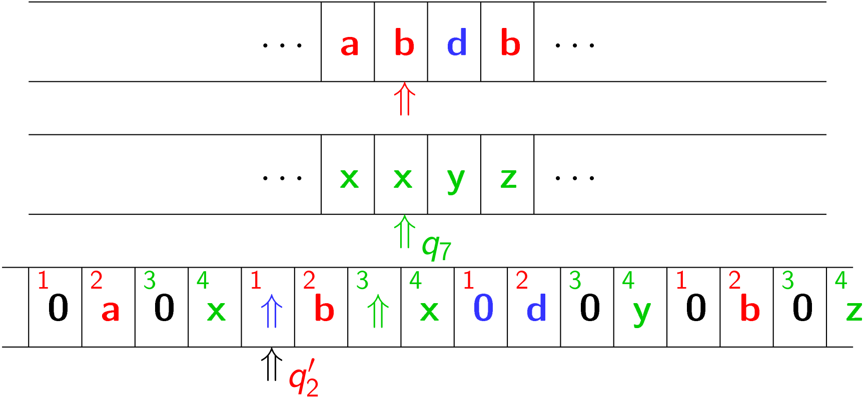
2) Рассмотрим 2-ленточную МТ $\mathcal{M}_2$ .




Получим одноленточную МТ $\mathcal{M}_1$ , которая может симулировать вычисления МТ $\mathcal{M}_2$ .

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

2) Рассмотрим 2-ленточную МТ $\mathcal{M}_2$ .




Получим одноленточную МТ $\mathcal{M}_1$ , которая может симулировать вычисления МТ $\mathcal{M}_2$ .

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

2) Рассмотрим 2-ленточную МТ $\mathcal{M}_2$ .

| | | | | | |
|---|---|---|---|---|---|
| $\dots$ | a | b | d | b | $\dots$ |
| | | $\Uparrow$ | | | |

| | | | | | |
|---|---|---|---|---|---|
| $\dots$ | x | x | y | z | $\dots$ |
| | | $\Uparrow$ $q_7$ | | | |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 0 | a | 0 | x | $\Uparrow$ | b | $\Uparrow$ | x | 0 | d | 0 | y | 0 | b | 0 | z |
| | | | | | | $\Uparrow$ $q_7$ | | | | | | | | | |

Получим одноленточную МТ $\mathcal{M}_1$ , которая может симулировать вычисления МТ $\mathcal{M}_2$ .

# Вариации машин Тьюринга

Как видно, вычислительные возможности всех n-ленточных МТ одинаковы для любых $n, n \geq 1$. Поэтому далее будем равноправно использовать все эти варианты МТ.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

Как видно, вычислительные возможности всех n-ленточных МТ одинаковы для любых $n, n \geq 1$. Поэтому далее будем равноправно использовать все эти варианты МТ.

Воспользуемся этой возможностью, чтобы доказать следующие теоремы, устанавливающие взаимосвязь между рекурсивными и рекурсивно перечислимыми языками.

**Теорема 4.4.** Язык $L$ является рекурсивным тогда и только тогда, когда оба языка $L$ и $\Sigma^* \setminus L$ являются рекурсивно перечислимыми.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

**Доказательство.** ($\Rightarrow$ ) Если $L = L(\mathcal{M})$ и $\mathcal{M}$ — тотальная МТ, то поменяв в ней местами допускающее и отвергающее состояния $q_0$ и $q_{-1}$, получим МТ, распознающую язык $\Sigma^* \setminus L$.

# Вариации машин Тьюринга

**Доказательство.** ($\Rightarrow$) Если $L = L(\mathcal{M})$ и $\mathcal{M}$ — тотальная МТ, то поменяв в ней местами допускающее и отвергающее состояния $q_0$ и $q_{-1}$, получим МТ, распознающую язык $\Sigma^* \setminus L$.

($\Leftarrow$) Предположим, что $L = L(\mathcal{M}_1)$ и $\Sigma^* \setminus L = L(\mathcal{M}_2)$. Построим 3-ленточную тотальную МТ $\mathcal{M}_0$, распознающую язык $L$. Эта машина работает так.

1. Копирует содержимое входной ленты на две рабочие ленты.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

**Доказательство.** ($\Rightarrow$ ) Если $L = L(\mathcal{M})$ и $\mathcal{M}$ — тотальная МТ, то поменяв в ней местами допускающее и отвергающее состояния $q_0$ и $q_{-1}$, получим МТ, распознающую язык $\Sigma^* \setminus L$.

($\Leftarrow$ ) Предположим, что $L = L(\mathcal{M}_1)$ и $\Sigma^* \setminus L = L(\mathcal{M}_2)$. Построим 3-ленточную тотальную МТ $\mathcal{M}_0$, распознающую язык $L$. Эта машина работает так.

1. Копирует содержимое входной ленты на две рабочие ленты.
2. Запускает на рабочих лентах МТ $\mathcal{M}_1$ и $\mathcal{M}_2$ и чередует шаги вычисления этих машин.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

**Доказательство.** ($\Rightarrow$) Если $L = L(\mathcal{M})$ и $\mathcal{M}$ — тотальная МТ, то поменяв в ней местами допускающее и отвергающее состояния $q_0$ и $q_{-1}$, получим МТ, распознающую язык $\Sigma^* \setminus L$.

($\Leftarrow$) Предположим, что $L = L(\mathcal{M}_1)$ и $\Sigma^* \setminus L = L(\mathcal{M}_2)$. Построим 3-ленточную тотальную МТ $\mathcal{M}_0$, распознающую язык $L$. Эта машина работает так.

1. Копирует содержимое входной ленты на две рабочие ленты.
2. Запускает на рабочих лентах МТ $\mathcal{M}_1$ и $\mathcal{M}_2$ и чередует шаги вычисления этих машин.
3. Если МТ $\mathcal{M}_1$ завершает свое вычисление в допускающем состоянии, то МТ $\mathcal{M}_0$ переходит в допускающее состояние. Если в допускающем состоянии завершает свое вычисление МТ $\mathcal{M}_2$, то МТ $\mathcal{M}_0$ переходит в отвергающее состояние.

QED

# Вариации машин Тьюринга

Пусть $\# \notin \Sigma$. Обозначим записью $\Sigma^*\#\Sigma^*$ множество слов $\{u\#v : u, v \in \Sigma^*\}$.

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

Пусть $\# \notin \Sigma$. Обозначим записью $\Sigma^*\#\Sigma^*$ множество слов $\{u\#v : u, v \in \Sigma^*\}$.

**Теорема 4.5.** Язык $L$ является рекурсивно перечислимым тогда и только тогда, когда существует такой рекурсивный язык $\widehat{L} \subseteq \Sigma^*\#\Sigma^*$, для которого верно равенство $L = \{w : \exists u \; w\#u \in \widehat{L}\}$.

**Доказательство.** ($\Rightarrow$ ) Пусть $L = L(\mathcal{M})$. Для каждого слова $w, w \in L,$ обозначим записью $t(w)$ число тактов вычисления МТ $\mathcal{M}$ при работе на входном слове $w$.

Рассмотрим язык $\widehat{L} = \{w\#u : w \in L, |u| > t(w)\}$.

**Доказательство.** ($\Rightarrow$) Пусть $L = L(\mathcal{M})$. Для каждого слова $w, w \in L,$ обозначим записью $t(w)$ число тактов вычисления МТ $\mathcal{M}$ при работе на входном слове $w$.

Рассмотрим язык $\widehat{L} = \{w\#u : w \in L, |u| > t(w)\}$.

Очевидно, что $L = \{w : \exists u \; w\#u \in \widehat{L}\}$.

**Доказательство.** ($\Rightarrow$ ) Пусть $L = L(\mathcal{M})$. Для каждого слова $w, w \in L,$ обозначим записью $t(w)$ число тактов вычисления МТ $\mathcal{M}$ при работе на входном слове $w$.

Рассмотрим язык $\widehat{L} = \{w\#u : w \in L, |u| > t(w)\}$.

Очевидно, что $L = \{w : \exists u \; w\#u \in \widehat{L}\}$.

3-ленточная МТ $\widehat{\mathcal{M}}$ работает так.

1. Получив входное слово $\widehat{w} = w\#u$, копирует слова $w$ и $u$ на две рабочие ленты.

**Доказательство.** ($\Rightarrow$ ) Пусть $L = L(\mathcal{M})$. Для каждого слова $w, w \in L,$ обозначим записью $t(w)$ число тактов вычисления МТ $\mathcal{M}$ при работе на входном слове $w$.

Рассмотрим язык $\widehat{L} = \{w\#u : w \in L, |u| > t(w)\}$.

Очевидно, что $L = \{w : \exists u \ w\#u \in \widehat{L}\}$.

3-ленточная МТ $\widehat{\mathcal{M}}$ работает так.

1. Получив входное слово $\widehat{w} = w\#u$, копирует слова $w$ и $u$ на две рабочие ленты.
2. Запускает на первой рабочей ленте МТ $\mathcal{M}$ и с каждым тактом ее работы стирает одну букву в слове $u$.

**Доказательство.** ($\Rightarrow$ ) Пусть $L = L(\mathcal{M})$. Для каждого слова $w$, $w \in L$, обозначим записью $t(w)$ число тактов вычисления МТ $\mathcal{M}$ при работе на входном слове $w$.

Рассмотрим язык $\widehat{L} = \{w\#u : w \in L, |u| > t(w)\}$.

Очевидно, что $L = \{w : \exists u \; w\#u \in \widehat{L}\}$.

3-ленточная МТ $\widehat{\mathcal{M}}$ работает так.

1. Получив входное слово $\widehat{w} = w\#u$, копирует слова $w$ и $u$ на две рабочие ленты.
2. Запускает на первой рабочей ленте МТ $\mathcal{M}$ и с каждым тактом ее работы стирает одну букву в слове $u$.
3. Если МТ $\mathcal{M}$ завершает вычисление на первой рабочей ленте в допускающем состоянии и при этом вторая рабочая лента непуста, то МТ $\widehat{\mathcal{M}}$ допускает слово $\widehat{w}$. Если МТ $\mathcal{M}$ завершает вычисление на первой рабочей ленте в отвергающем состоянии или вторая лента опустошается, то МТ $\widehat{\mathcal{M}}$ отвергает слово $\widehat{w}$.

Нетрудно видеть, что $\widehat{\mathcal{M}}$ — тотальная МТ, и $L(\widehat{\mathcal{M}}) = \widehat{L}$.

**Доказательство.** ($\Leftarrow$) Пусть $\widehat{L} = L(\widehat{\mathcal{M}})$.

3-ленточная МТ $\mathcal{M}$, получив входное слово $w$, выполняет бесконечный цикл.

**Доказательство.** ($\Leftarrow$ ) Пусть $\widehat{L} = L(\widehat{\mathcal{M}})$ .

3-ленточная МТ $\mathcal{M}$ , получив входное слово $w$ , выполняет бесконечный цикл.

1. На первой рабочей ленте $\mathcal{M}$ порождает все слова в алфавите $\Sigma$ поочередно в лексикографическом порядке, по одному слову $u$ на каждой итерации цикла.

**Доказательство.** ($\Leftarrow$ ) Пусть $\widehat{L} = L(\widehat{\mathcal{M}})$.

3-ленточная МТ $\mathcal{M}$, получив входное слово $w$, выполняет бесконечный цикл.

1. На первой рабочей ленте $\mathcal{M}$ порождает все слова в алфавите $\Sigma$ поочередно в лексикографическом порядке, по одному слову $u$ на каждой итерации цикла.
2. После того, как на очередной итерации цикла МТ $\mathcal{M}$ сгенерировала на рабочей ленте слово $u$, она формирует на второй рабочей ленте слово $\widehat{w} = w\#u$ и запускает на этом слове МТ $\widehat{\mathcal{M}}$. Если $\widehat{\mathcal{M}}$ допускает слово $\widehat{w}$, то МТ $\mathcal{M}$ завершает работу и переходит в допускающее состояние. Если $\widehat{\mathcal{M}}$ отвергает слово $\widehat{w}$, то МТ $\mathcal{M}$ продолжает выполнение цикла и порождает на первой рабочей ленте очередное слово.

Нетрудно видеть, что $L(\mathcal{M}) = \{w : \exists u \; w\#u \in \widehat{L}\} = L$.

QED

# ВАРИАЦИИ МАШИН ТЬЮРИНГА

**Задача 4.1.** Докажите, что вычислительные возможности машины Тьюринга не изменяются, если ее вместо бесконечной ленты снабдить

1. двумя бесконечными стеками (магазинами);
2. бесконечной плоскостью, разбитой на ячейки, подобно ленте.

**Задача 4.2.** Докажите, что вычислительные возможности машины Тьюринга не зависят от числа символов рабочего алфавита $\Gamma$.

# Вариации машин Тьюринга

**Задача 4.3.** Докажите, что вычислительные возможности машины Тьюринга не изменяются, если вместо функции переходов она использует отношение переходов, позволяющее в некоторых конфигурациях недетерминированно выбирать для выполнения одну из нескольких команд.

**Задача 4.4. (Трудная)** Докажите, что вычислительные возможности машины Тьюринга не изменяются, если вместо бесконечной ленты она будет работать с очередью, считывая буквы из начала очереди и записывая буквы в конец очереди.

# Свойства замкнутости

**Теорема 4.6.** Класс рекурсивных языков замкнут относительно операций объединения, пересечения и дополнения.

**Доказательство.** Пусть рекурсивные языки $L_1$ и $L_2$ распознаются тотальными 1-летночными МТ $\mathcal{M}_1$ и $\mathcal{M}_2$. Построим тотальную 2-ленточную МТ $\mathcal{M}_{1\cap 2}$, распознающую язык $L_1 \cap L_2$.

# Свойства замкнутости

**Теорема 4.6.** Класс рекурсивных языков замкнут относительно операций объединения, пересечения и дополнения.

**Доказательство.** Пусть рекурсивные языки $L_1$ и $L_2$ распознаются тотальными 1-летночными МТ $\mathcal{M}_1$ и $\mathcal{M}_2$. Построим тотальную 2-ленточную МТ $\mathcal{M}_{1\cap 2}$, распознающую язык $L_1 \cap L_2$.

МТ $\mathcal{M}_{1\cap 2}$, получив входное слово, копирует его на рабочую ленту и запускает на входной ленте МТ $\mathcal{M}_1$. Если эта МТ достигает допускающего состояния, то на рабочей ленте запускается МТ $\mathcal{M}_2$. Если и она достигант допускающего состояния, то МТ $\mathcal{M}_{1\cap 2}$ допускает входное слово.

# Свойства замкнутости

**Теорема 4.6.** Класс рекурсивных языков замкнут относительно операций объединения, пересечения и дополнения.

**Доказательство.** Пусть рекурсивные языки $L_1$ и $L_2$ распознаются тотальными 1-летночными МТ $\mathcal{M}_1$ и $\mathcal{M}_2$. Построим тотальную 2-ленточную МТ $\mathcal{M}_{1\cap 2}$, распознающую язык $L_1 \cap L_2$.

МТ $\mathcal{M}_{1\cap 2}$, получив входное слово, копирует его на рабочую ленту и запускает на входной ленте МТ $\mathcal{M}_1$. Если эта МТ достигает допускающего состояния, то на рабочей ленте запускается МТ $\mathcal{M}_2$. Если и она достигант допускающего состояния, то МТ $\mathcal{M}_{1\cap 2}$ допускает входное слово.

Покажите самостоятельно, как построить МТ, допускающие языки $L_1 \cup L_2$ и $\Sigma^* \setminus L_1$ QED

# Свойства замкнутости

**Теорема 4.7.** Класс рекурсивно перечислимых языков замкнут относительно операций объединения и пересечения.

**Доказательство.** Пусть рекурсивно перечислимые языки $L_1$ и $L_2$ распознаются 1-летночными МТ $\mathcal{M}_1$ и $\mathcal{M}_2$. Построим 2-ленточную МТ $\mathcal{M}_{1\cup 2}$, распознающую язык $L_1 \cup L_2$.

# Свойства замкнутости

**Теорема 4.7.** Класс рекурсивно перечислимых языков замкнут относительно операций объединения и пересечения.

**Доказательство.** Пусть рекурсивно перечислимые языки $L_1$ и $L_2$ распознаются 1-летночными МТ $\mathcal{M}_1$ и $\mathcal{M}_2$. Построим 2-ленточную МТ $\mathcal{M}_{1\cup 2}$, распознающую язык $L_1 \cup L_2$.

МТ $\mathcal{M}_{1\cup 2}$, получив входное слово, копирует его на рабочую ленту и запускает на обеих лентах МТ $\mathcal{M}_1$ и $\mathcal{M}_2$, чередуя шаги их вычислений. Если хотя бы одна из этих МТ достигает допускающего состояния, то МТ $\mathcal{M}_{1\cap 2}$ допускает входное слово.

# Свойства замкнутости

**Теорема 4.7.** Класс рекурсивно перечислимых языков замкнут относительно операций объединения и пересечения.

**Доказательство.** Пусть рекурсивно перечислимые языки $L_1$ и $L_2$ распознаются 1-летночными МТ $\mathcal{M}_1$ и $\mathcal{M}_2$. Построим 2-ленточную МТ $\mathcal{M}_{1\cup 2}$, распознающую язык $L_1 \cup L_2$.

МТ $\mathcal{M}_{1\cup 2}$, получив входное слово, копирует его на рабочую ленту и запускает на обеих лентах МТ $\mathcal{M}_1$ и $\mathcal{M}_2$, чередуя шаги их вычислений. Если хотя бы одна из этих МТ достигает допускающего состояния, то МТ $\mathcal{M}_{1\cap 2}$ допускает входное слово.

Покажите самостоятельно, как построить МТ, допускающую язык $L_1 \cap L_2$ QED

# Свойства замкнутости

**Теорема 4.7.** Класс рекурсивно перечислимых языков замкнут относительно операций объединения и пересечения.

**Доказательство.** Пусть рекурсивно перечислимые языки $L_1$ и $L_2$ распознаются 1-летночными МТ $\mathcal{M}_1$ и $\mathcal{M}_2$. Построим 2-ленточную МТ $\mathcal{M}_{1\cup 2}$, распознающую язык $L_1 \cup L_2$.

МТ $\mathcal{M}_{1\cup 2}$, получив входное слово, копирует его на рабочую ленту и запускает на обеих лентах МТ $\mathcal{M}_1$ и $\mathcal{M}_2$, чередуя шаги их вычислений. Если хотя бы одна из этих МТ достигает допускающего состояния, то МТ $\mathcal{M}_{1\cap 2}$ допускает входное слово.

Покажите самостоятельно, как построить МТ, допускающую язык $L_1 \cap L_2$ QED

Вопрос о замкнутости класса рекурсивно перечислимых языков относительно дополнения равносилен вопросу о совпадении этого класса с классом рекурсивных языков.

# Функции, вычислимые по Тьюрингу

Многоленточные МТ можно использовать не только для распознавания слов, но и для вычисления одних слов из других.

Для этой цели выделяется отдельная лента — лента выхода . Получив слово на входной ленте, МТ проводит вычисление, используя рабочие ленты, записывает результат (слово в выходном алфавите $\Delta$ ) на ленте выхода и останавливается в допускающем состоянии (или зацикливается).

Словарные функции $F : \Sigma^* \to \Delta^*$ , вычислимые таким образом на МТ, называются вычислимыми по Тьюрингу (коротко, Т-функциями).

# Функции, вычислимые по Тьюрингу

Поскольку каждое слово $w = a_{j_1} a_{j_2} \dots a_{j_m}$ в алфавите $a_1, \dots, a_k$ представляет запись натурального числа $n_w = \sum_{i=1}^{m} j_i \cdot k^{m-i}$, словарные функции $F : \Sigma^* \to \Delta^*$ можно истолковывать, как арифметические функции.

Известна следующая

# Функции, вычислимые по Тьюрингу

Поскольку каждое слово $w = a_{j_1} a_{j_2} \dots a_{j_m}$ в алфавите $a_1, \dots, a_k$ представляет запись натурального числа $n_w = \sum_{i=1}^{m} j_i \cdot k^{m-i}$, словарные функции $F : \Sigma^* \to \Delta^*$ можно истолковывать, как арифметические функции.

Известна следующая

**Теорема.** [С. Клини] Класс арифметических Т-функций совпадает с классом частично рекурсивных функций.

# Функции, вычислимые по Тьюрингу

Класс рекурсивных и рекурсивно перечислимых языков можно охарактеризовать в терминах Т-функций.

**Теорема 4.8.** Язык $L, L \subseteq \Sigma^*$, является рекурсивным тогда и только тогда, когда существует Т-функция

$$F(w) = \begin{cases} 1, & \text{если } w \in L, \\ 0, & \text{если } w \notin L, \end{cases}$$

**Доказательство.** Очевидно.

# Функции, вычислимые по Тьюрингу

**Теорема 4.9.** Язык $L, L \subseteq \Sigma^*,$ является рекурсивно перечислимым тогда и только тогда, когда существует Т-функция

$$H(w) = \begin{cases} 1, \text{ если } w \in L, \\ \text{неопределено, если } w \notin L, \end{cases}$$

**Доказательство.** Очевидно.

# ФУНКЦИИ, ВЫЧИСЛИМЫЕ ПО ТЬЮРИНГУ

**Теорема 4.9.** Язык $L, L \subseteq \Sigma^*$, является рекурсивно перечислимым тогда и только тогда, когда существует Т-функция

$$H(w) = \begin{cases} 1, & \text{если } w \in L, \\ \text{неопределено}, & \text{если } w \notin L, \end{cases}$$

**Доказательство.** Очевидно.

Но есть и более интересное описание рекурсивно перечислимых языков, объясняющее происхождение этого термина.

## Функции, вычислимые по Тьюрингу

**Теорема 4.10.** Язык $L, L \subseteq \Sigma^*, L \neq \varnothing,$ является рекурсивно перечислимым тогда и только тогда, когда существует такая всюду определенная Т-функция $E : \mathbb{N} \to \Sigma^*$, областью значений которой является язык $L$.

Теорема утверждает, что рекурсивная функция $E$ способна перечислить все слова языка $L$.

# Функции, вычислимые по Тьюрингу

**Доказательство.** ($\Leftarrow$) Пусть $L = E(\mathbb{N})$.

Следующая МТ $\mathcal{M}$ распознает язык $L$.

Для проверки включения $w \in L$ она запускает МТ, вычисляющую функцию $E$, последовательно на всех натуральных числах, до тех пор пока не будет вычислено выходное слово $w$.

Если это случится, то $\mathcal{M}$ переходит в допускающее состояние.

# Функции, вычислимые по Тьюрингу

**Доказательство.** ($\Rightarrow$ )
Предположим, что МТ $\mathcal{M}$ распознает язык $L$ .

Поскольку язык $L$ непуст, существует входное слово $w_0, w_0 \in L$ .

Расположим все входные слова из множества $\Sigma^*$ в лексикографическом порядке: $u_1, u_2, u_3, \ldots$ .

**Доказательство.** ($\Rightarrow$)
Предположим, что МТ $\mathcal{M}$ распознает язык $L$.

Поскольку язык $L$ непуст, существует входное слово $w_0$, $w_0 \in L$.

Расположим все входные слова из множества $\Sigma^*$ в лексикографическом порядке: $u_1, u_2, u_3, \ldots$ .

А теперь приведем рекурсивное описание функции $E$, перечисляющей все слова языка $L$.

# ФУНКЦИИ, ВЫЧИСЛИМЫЕ ПО ТЬЮРИНГУ

0). Положим $E(0) = w_0$ .

# Функции, вычислимые по Тьюрингу

0). Положим $E(0) = w_0$ .

n $\rightarrow$ n+1). Чтобы вычислить $E(n+1)$ нужно

1. Вычислить список входных слов
   $W_n = \{E(0), E(1), \ldots, E(n)\}$ ;

# ФУНКЦИИ, ВЫЧИСЛИМЫЕ ПО ТЬЮРИНГУ

0). Положим $E(0) = w_0$ .

n $\to$ n+1). Чтобы вычислить $E(n+1)$ нужно

1. Вычислить список входных слов $W_n = \{E(0), E(1), \ldots, E(n)\}$ ;
2. Сформировать список слов $U_n = \{u_0, u_1, \ldots, u_n, u_{n+1}\} \setminus W_n$ ;

# ФУНКЦИИ, ВЫЧИСЛИМЫЕ ПО ТЬЮРИНГУ

0). Положим $E(0) = w_0$ .

n $\to$ n+1). Чтобы вычислить $E(n+1)$ нужно

1. Вычислить список входных слов
   $W_n = \{E(0), E(1), \ldots, E(n)\}$ ;
2. Сформировать список слов $U_n = \{u_0, u_1, \ldots, u_n, u_{n+1}\} \setminus W_n$ ;
3. Применить МТ $\mathcal{M}$ к каждому слову из списка $U_n$ ,
   ограничив время ее работы $n+1$ тактами;

# ФУНКЦИИ, ВЫЧИСЛИМЫЕ ПО ТЬЮРИНГУ

0). Положим $E(0) = w_0$ .

n $\rightarrow$ n+1). Чтобы вычислить $E(n+1)$ нужно

1. Вычислить список входных слов
   $W_n = \{E(0), E(1), \ldots, E(n)\}$ ;
2. Сформировать список слов $U_n = \{u_0, u_1, \ldots, u_n, u_{n+1}\} \setminus W_n$ ;
3. Применить МТ $\mathcal{M}$ к каждому слову из списка $U_n$ , ограничив время ее работы $n+1$ тактами;
4. Если МТ $\mathcal{M}$ при работе с каждым из слов списка $U_n$ не достигает допускающего состояния за $n+1$ или менее тактов работы, то положить $E(n+1) = w_0$ ;

# ФУНКЦИИ, ВЫЧИСЛИМЫЕ ПО ТЬЮРИНГУ

0). Положим $E(0) = w_0$ .

n $\to$ n+1). Чтобы вычислить $E(n+1)$ нужно

1. Вычислить список входных слов $W_n = \{E(0), E(1), \dots, E(n)\}$ ;
2. Сформировать список слов $U_n = \{u_0, u_1, \dots, u_n, u_{n+1}\} \setminus W_n$ ;
3. Применить МТ $\mathcal{M}$ к каждому слову из списка $U_n$ , ограничив время ее работы $n+1$ тактами;
4. Если МТ $\mathcal{M}$ при работе с каждым из слов списка $U_n$ не достигает допускающего состояния за $n+1$ или менее тактов работы, то положить $E(n+1) = w_0$ ;
5. В противном случае выбрать слово $u$ с наименьшим порядковым номером среди тех слов списка $U_n$ , при работе с которыми МТ $\mathcal{M}$ достигла допускающего состояния не более чем за $n+1$ тактов работы, и положить $E(n+1) = u$ .

## Функции, вычислимые по Тьюрингу

Из приведенного описания видно, что

- $E(x)$ — всюду определенная Т-вычислимая функция;

# Функции, вычислимые по Тьюрингу

Из приведенного описания видно, что

- $E(x)$ — всюду определенная Т-вычислимая функция;
- значениями функции $E(x)$ являются только слова языка $L = L(\mathcal{M})$;

# ФУНКЦИИ, ВЫЧИСЛИМЫЕ ПО ТЬЮРИНГУ

Из приведенного описания видно, что

- $E(x)$ — всюду определенная Т-вычислимая функция;
- значениями функции $E(x)$ являются только слова языка $L = L(\mathcal{M})$;
- если входное слово $u$ имеет порядковый номер $k$ и допускается МТ $\mathcal{M}$ за $m$ тактов работы, то $E(n) = u$ для некоторого $n$, $0 \leq n \leq m + k$.

Таким образом, всякий рекурсивно перечислимый язык генерируется некоторой всюду определенной Т-вычислимой (т.е. рекурсивной) функцией. QED

# Функции, вычислимые по Тьюрингу

**Задача 4.5.** Докажите, что бесконечный язык $L, L \subseteq \Sigma^*, L \neq \varnothing,$ является рекурсивным тогда и только тогда, когда существует всюду определенная Т-функция $E : \mathbb{N} \to \Sigma^*$, перечисляющая слова этого языка в лексикографическом порядке.

**Задача 4.6.** Докажите, что язык $L, L \subseteq \Sigma^*,$ является рекурсивно перечислимым, если существует **недетерминированная** МТ, которая, которая получив на входе пустую ленту, может напечатать на выходной ленте все слова языка $L, L \subseteq \Sigma^*,$ и только эти слова.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Математические задачи можно разделить на два класса:

- индивидуальные задачи : например, решить уравнение $x^4 + 2x^2 + 1 = 0$ ;
- массовые проблемы : например, решить любое уравнение вида $ax^4 + bx^3 + cx^2 + dx + e = 0$.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Математические задачи можно разделить на два класса:

- индивидуальные задачи : например, решить уравнение $x^4 + 2x^2 + 1 = 0$ ;
- массовые проблемы : например, решить любое уравнение вида $ax^4 + bx^3 + cx^2 + dx + e = 0$.

Чтобы продемонстрировать способность решать индивидуальную задачу, Вам достаточно привести ее правильное решение, и объяснить, как проверить его правильность.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Математические задачи можно разделить на два класса:

- индивидуальные задачи: например, решить уравнение $x^4 + 2x^2 + 1 = 0$;
- массовые проблемы: например, решить любое уравнение вида $ax^4 + bx^3 + cx^2 + dx + e = 0$.

Чтобы продемонстрировать способность решать индивидуальную задачу, Вам достаточно привести ее правильное решение, и объяснить, как проверить его правильность.

А как подтвердить умение решать массовую проблему?

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

До середины XX века математики не знали средств для формального определения понятия «способ решения задачи».
И поэтому они не могли доказывать утверждения о том, что некоторые массовые проблемы не имеют способа их решения.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

До середины XX века математики не знали средств для формального определения понятия «способ решения задачи».
И поэтому они не могли доказывать утверждения о том, что некоторые массовые проблемы не имеют способа их решения.

С появлением формальных определений понятия «алгоритма» математики пришли к соглашению о том, что способом решения задачи следует считать формальный алгоритм ее решения.
И тогда появилась возможность формализовать понятие «массовой алгоритмической проблемы».

## МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Мы будем рассматривать индивидуальные задачи распознавания, решениями которых служат ответы «да» или «нет». Например,

$P_1$: Являются ли два заданных графа $G_1$ и $G_2$ изоморфными?

$P_2$: Имеет ли заданное алгебраическое уравнение $E(x)=0$ решение, меньшее заданного числа $c$?

$P_3$: Выполнима ли заданная формула логики предикатов $\Phi$?

$P_4$: Верно ли, что заданная МТ $\mathcal{M}$ завершает вычисления на всех начальных конфигурациях?

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Каждая индивидуальная задача полностью характеризуется набором своих специфических параметров:

- $P_1 = \langle G_1, G_2 \rangle$ ,
- $P_2 = \langle E(x), c \rangle$ ,
- $P_3 = \langle \Phi \rangle$ ,
- $P_4 = \langle \mathcal{M} \rangle$ ,

который может быть представлен (записан, закодирован) словом в некотором конечном алфавите.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Массовая проблема — это совокупность всех индивидуальных задач определенного типа, в которой выделено множество индивидуальных задач (вопросов), правильным ответом для которых является утвердительный ответ «да»:

- $\mathcal{P}_1 = \{\langle G_1, G_2\rangle : G_1 \text{ и } G_2 \text{ изоморфны}\}$,
- $\mathcal{P}_2 = \{\langle E, c\rangle : \exists x(E(x) = 0 \wedge x < c)\}$,
- $\mathcal{P}_3 = \{\langle \Phi\rangle : \exists I(I \models \Phi)\}$,
- $\mathcal{P}_4 = \{\langle \mathcal{M}\rangle : \forall \alpha(\mathcal{M}(\alpha) \text{ конечно})\}$.

Таким образом, каждая массовая проблема представляет собой некоторый язык в конечном алфавите.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Решить массовую проблему $\mathcal{P}$ означает найти МТ (алгоритм) $\mathcal{M}$, который для любой индивидуальной задачи $w$:

- переходит в допускающее состояние, если правильным ответом для задачи $w$ является утвердительный ответ, т.е. $w \in \mathcal{P}$;
- переходит в отвергающее состояние, если задача $w$ имеет отрицательный ответ (или некорректно сформулирована), т.е. $w \notin \mathcal{P}$;.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Решить массовую проблему $\mathcal{P}$ означает найти МТ (алгоритм) $\mathcal{M}$, который для любой индивидуальной задачи $w$:

- переходит в допускающее состояние, если правильным ответом для задачи $w$ является утвердительный ответ, т.е. $w \in \mathcal{P}$;
- переходит в отвергающее состояние, если задача $w$ имеет отрицательный ответ (или некорректно сформулирована), т.е. $w \notin \mathcal{P}$;.

Таким образом, массовая проблема $\mathcal{P}$ считается алгоритмически разрешимой, если $\mathcal{P}$ — рекурсивный язык.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

А каково положение рекурсивно перечислимых языков среди массовых алгоритмических проблем?

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

А каково положение рекурсивно перечислимых языков среди массовых алгоритмических проблем?

Согласно теореме 4.5, язык $L$ рекурсивно перечислим тогда и только тогда, когда $L = \{w : \exists u : w\#u \in \widehat{L}\}$ для некоторого рекурсивного языка $\widehat{L} \subseteq \Sigma^*\#\Sigma^*$ .

А что представляет собой язык $\widehat{L}$?

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Язык $\widehat{L}$ состоит из слов $w\#u$, в которых

- $w$ — это индивидуальная задача (уравнение $E(x) = 0$, пара графов $(G_1, G_2)$ и др.), имеющая положительное решение,
- $u$ — это решение задачи $w$, т.е. «сертификат» (корень уравнения, биекция изоморфизма и др), удостоверяющий правильность ответа.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Язык $\widehat{L}$ состоит из слов $w\#u$, в которых

- $w$ — это индивидуальная задача (уравнение $E(x) = 0$, пара графов $(G_1, G_2)$ и др.), имеющая положительное решение,
- $u$ — это решение задачи $w$, т.е. «сертификат» (корень уравнения, биекция изоморфизма и др), удостоверяющий правильность ответа.

Рекурсивность языка $\widehat{L}$ означает, что есть алгоритм, проверяющий правильность решения $u$ каждой индивидуальной задачи $w$ массовой проблемы $L$!

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Поэтому вопрос о том, совпадают ли классы рекурсивных и рекурсивно перечислимых языков можно истолковать так:

**Верно ли, что каждая проблема, для которой есть**
**алгоритм проверки правильности решения,**
**имеет также и**
**алгоритм получения решения?**

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Этот вопрос можно выразить еще проще.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Этот вопрос можно выразить еще проще.

**Хороший студент** — тот, кто умеет правильно решать задачи.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Этот вопрос можно выразить еще проще.

**Хороший студент** — тот, кто умеет правильно решать задачи.

**Хороший экзаменатор** — тот, кто безошибочно проверяет решения задач.

# МАССОВЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Этот вопрос можно выразить еще проще.

**Хороший студент** — тот, кто умеет правильно решать задачи.

**Хороший экзаменатор** — тот, кто безошибочно проверяет решения задач.

**Верно ли, что хороший экзаменатор был хорошим студентом?**

КОНЕЦ ЛЕКЦИИ 4