Самоприменимые и несамоприменимые МТ. Доказательство существования алгоритмически неразрешимых задач. Тезис Черча-Тьюринга.

- Машина называется самоприменимой, если после начала работы в указанной конфигурации она через конечное число шагов попадет в заключительное состояние q_0,
- в противном случае машина называется несамоприменимой.
- Пример: Машина $q_1 1 \to 1Hq_0$ $q_1 0 \to 0Hq_0$ является самоприменимой, поскольку после выполнения одной команды машина попадает в заключительное состояние независимо от того, что было записано на ленте (т.е. в том числе и если был записан шифр этой машины
- Пример: Программа машины $q_1 1 \to Hq_1$ $q_1 0 \to 0Hq_1$ не содержит состояния q_0, поэтому машина не может попасть в это состояние и является несамоприменимой

Проблема самоприменимости

- Пусть арифметическая функция *p(x)* определена следующим образом

$$p(x)=\begin{cases}1, \text{если } x \text{ шифр самоприменимой машины,}\\ 0, \text{если } x \text{ шифр несамоприменимой машины}\end{cases}$$

- Существует ли машина Тьюринга *T* в алфавите $\{a_0, 0, 1\}$, которая правильно вычисляет функцию *p(x)*?

- Будем считать, что машина будет оставлять на ленте одну единицу в случае, если слово на ленте в начальной конфигурации было шифром некоторой самоприменимой машины,
- и будет оставлять 0, если слово на ленте в начальной конфигурации было шифром некоторой несамоприменимой машины.

- Если машина Тьюринга существует для решения задачи, то такая задача называется алгоритмически разрешимой,
- впротивном случае задача алгоритмически неразрешима.

1. Счётность множества алгоритмов

Представим все возможные алгоритмы (программы) в виде конечных строк над конечным алфавитом (например, бинарные строки, исходный код на Python и т. д.).
Таких строк счётное число — их можно перенумеровать:
A1,A2,A3,…A1,A2,A3,…

---

2. Несчётность множества задач (функций)

Рассмотрим множество всех функций вида:

f:N→{0,1}f:N→{0,1}

Таких функций континуум (2ℵ02ℵ0), что **несчётно** (по теореме Кантора).

Каждая вычислимая функция соответствует какому-то алгоритму. Но алгоритмов — счётное число.
Следовательно, **большинство функций невычислимы**.

---

3. Явный пример: проблема остановки

Проблема остановки (Halting Problem) — классический пример алгоритмически неразрешимой задачи.

Формулировка:
Даны программа PP и её вход xx. Определить, останавливается ли PP на входе xx.

Доказательство неразрешимости (Тьюринг, 1936):

Предположим, существует программа HH, которая решает проблему остановки:

H(P,x)={1,если P останавливается на x0,иначеH(P,x)={1,0,
если P останавливается на хиначе

Построим программу DD (диагональная конструкция):

text

D(P):
если H(P, P) == 1: // спрашиваем, останавливается ли P на своём собственном коде
войти в бесконечный цикл
иначе:
остановиться

Теперь рассмотрим, что происходит, когда DD получает на вход свой собственный код DD:

- Если H(D,D)=1H(D,D)=1 (то есть HH говорит, что DD останавливается на DD), то DD по построению уходит в бесконечный цикл → **противоречие**.
- Если H(D,D)=0H(D,D)=0 (то есть HH говорит, что DD не останавливается на DD), то DD по построению останавливается → **противоречие**.

Следовательно, такой программы HH существовать не может → проблема остановки **неразрешима**.

---

4. Следствия и другие неразрешимые задачи

Из неразрешимости проблемы остановки можно вывести неразрешимость многих других задач:

- Проблема эквивалентности программ (определить, вычисляют ли две программы одну и ту же функцию)
- Проблема тотальности (определить, останавливается ли программа на всех входах)
- Десятая проблема Гильберта (разрешимость диофантовых уравнений) — доказана Ю. В. Матиясевичем в 1970 г.

---

5. Итог

Доказательство существования алгоритмически неразрешимых задач состоит из двух частей:

1. **Теоретико-множественный аргумент**: алгоритмов счётно, а функций несчётно → почти все функции невычислимы.
2. **Конкретный пример**: проблема остановки неразрешима (доказано от противного через диагонализацию).

Тезис Чёрча-Тьюринга:

Для любой интуитивно вычислимой функции существует вычисляющая её значения машина Тьюринга