Алгоритмически неразрешимые проблемы

В теории алгоритмов известны задачи, о которых ***доказано***, что для их решения не существует алгоритма.

Такие задачи называются ***алгоритмически неразрешимыми***.

ТЕОРЕМА (проблема “останова”):

Не существует алгоритма (машины Тьюринга), позволяющего по описанию произвольного алгоритма и его исходных данных (и алгоритм и данные заданы символами на ленте машины Тьюринга) определить, **останавливается ли** этот алгоритм на этих данных или работает бесконечно.

Иными словами: проблема «останова», т.е. определения по записи произвольного алгоритма , а также по записи произвольного "входа", остановится ли вычислительное устройство, действующее в соответствие с данным алгоритмом и обрабатывающее данный "вход", или же оно будет работать бесконечно долго, является алгоритмически неразрешимой.

ПОНЯТИЕ САМОПРИМЕНИМОСТИ АЛГОРИТМА :

Алгоритм называется **самоприменимым**, если он эффективно перерабатывает текст, соответствующий его собственной записи, в некоторый результат за конечное число шагов.

В противном случае - если алгоритм не останавливается и продолжает работать бесконечно долго, он называется **несамоприменимым**.

ТЕОРЕМА (проблема распознавания самоприменимости):

Проблема распознавания самоприменимости алгоритмически неразрешима.

Проблема распознавания самоприменимости является частным случаем проблемы останова. Алгоритмическая неразрешимость более общей проблемы будет очевидна.

ДОКАЗАТЕЛЬСТВО:

От противного

Пусть такая машина существует.

Назовем ее А.

Тогда А

- любой самоприменимый шифр перерабатывает в ‘1’
  (значит машина самоприменима)
- любой несамоприменимый в ‘0’
  (машина несамоприменима)

Если существует такая машина А,

то существует и такая машина В, что:

- несамоприменимые машины перерабатывает в ‘1’
- самоприменимые не перерабатывает (работает бесконечно)

Рассмотрим два случая:

1. В самоприменима

1. В несамоприменима

1. Если B самоприменима, то она, будучи применена к своему шрифту B*, будет работает бесконечно.

   Но это означает, что данный алгоритм несамоприменим. Противоречие.

2. Если B несамоприменима, то она, будучи применена к своему шифру B*, остановится с 1.
   Применимость B к B* означает, что B самоприменима.
   Противоречие.

Следовательно, таких машин A и B не существует.

Теорема доказана.

ПРИМЕРЫ НЕРАЗРЕШИМЫХ ПРОБЛЕМ:

- **Десятая проблема Гильберта**

Пусть задан многочлен n-ой степени с целыми коэффициентами – P, существует ли алгоритм, который определяет, имеет ли уравнение P=0 решение в целых числах?

Ю.В. Матиясевич показал, что такого алгоритма **не существует**, т.е. отсутствует общий метод определения целых корней уравнения P=0 по его целочисленным коэффициентам.

❖ Проблема тотальности

По произвольному заданному алгоритму определить, будет ли он останавливаться на всех возможных наборах исходных данных.

Другая формулировка этой задачи – определить, является ли алгоритм P всюду определённым или частично определенным?

- ## Проблема “останова” и сводящиеся к ней задачи
- **ТЕОРЕМА:**

Задача о печатании данного знака машины на чистой ленте точно один раз алгоритмически неразрешима.

ДОКАЗАТЕЛЬСТВО:

Пусть дана машина Т, которая работает на чистой ленте.

1. Преобразуем машину Т в машину D
   1.1 если в алфавите машины Т нет “0”, то D совпадает с Т
   1.2 если “0” есть, то заменяем в алфавите D “0” новым знаком, который не содержался в ее алфавите.
2. Построим машину Е такую, которая работает также, как и машина Т, но после остановки машины Т она ставит “0” и тоже останавливается.
3. Вывод: “0” печатается один раз, если машина Тьюринга останавливается на чистой ленте.

Задача о печатании “0” равносильна задаче об остановке машины на чистой ленте.

❖ ТЕОРЕМА:

Задача о печатании данного знака машины на чистой ленте бесконечное количество раз алгоритмически неразрешима

Доказательство:

Возьмем знак «0». Пусть машина Тьюринга T работает на чистой ленте.

1. Преобразим ее в новую машину Тьюринга D.

1.1 Если T не содержит в своем алфавите «0», то D совпадает с T.

1.2 Если T имеет этот знак в своем алфавите, то в алфавите машины D «0» будет заменен любым знаком, ранее не входящим в алфавит машины.

D остановится тогда, когда остановится T.

2. Построим машину E, которая работает как D вплоть до ее остановки. После чего E переходит в состояние A и печатает «0» бесконечно много раз ($A\lambda \rightarrow 0RA$).

3. Получим, что символ «0» печатается бесконечно много раз в том и только в том случае, если машина T останавливается на чистой ленте. Значит задача о печатании бесконечно большого числа нулей равносильна задаче об остановке машины на чистой ленте.

Поскольку задача останова алгоритмически неразрешима, то и задача о печатании символа бесконечно много раз тоже *неразрешима*.

- **ТЕОРЕМА:**

  Проблема распознавания эквивалентности машин Тьюринга является алгоритмически неразрешимой.

  По двум произвольным машинам Тьюринга определить, будут ли они выдавать одинаковые выходные результаты на всех исходных данных.

Примеры задач, для которых не найден алгоритм решения

1. Вычисление совершенных чисел

Совершенные числа – это числа, которые равны сумме своих делителей, например: 28 = 1+2+4+7+14.

Определим функцию S(n) = n-ое по счёту совершенное число и поставим задачу вычисления S(n) по произвольно заданному n. **Не найден общий метод вычисления совершенных чисел**, мы даже не знаем, множество совершенных чисел конечно или счетно.

2. Распределение девяток в записи числа π(пи)

Определим функцию f(n) = i,

где n – количество девяток подряд в десятичной записи числа π

,

а i – номер самой левой девятки из n девяток подряд:

π =3,141592… f(1) = 5.

Задача состоит в вычислении функции f(n) для произвольно заданного n.

Поскольку число π является иррациональным и трансцендентным, то мы не знаем никакой информации о распределении девяток (равно как и любых других цифр) в десятичной записи числа π . Вычисление f(n) связано с вычислением последующих цифр в разложении π , до тех пор, пока мы не обнаружим n девяток подряд, однако у нас нет общего метода вычисления f(n), поэтому для некоторых n вычисления могут продолжаться бесконечно – мы даже не знаем в принципе (по природе числа π) существует ли решение для всех n.

Вопросы для самопроверки:

1. Какие задачи называются алгоритмически неразрешимыми?
2. Сформулируйте проблему тотальности.
3. Какой алгоритм называется самоприменимым/несамоприменимым?
4. В чем заключается проблема останова?
5. Какой способ доказательства неразрешимости задачи является самым распространенным?
6. Является ли задача, для которой не найден алгоритм решения неразрешимой?
7. Сформулируйте проблему распознавания эквивалентности.