**УДК 510.643**

# Неразрешимость и труднорешаемые проблемы

***А.Н. Гамова***

*Саратовский национальный исследовательский государственный университет имени Н.Г. Чернышевского, г. Саратов*

Общее понятие алгоритма первично и не определяемо, а может быть только понято через его свойства, подобно понятию множества. Через вычислительные модели (машина Тьюринга, Колмогорова, нормальные алгоритмы Маркова) оно уточняется. То же относится к понятию исчисления, которое уточняется в конкретных дедуктивных системах (логические исчисления, формальная арифметика). Между алгоритмами и исчислениями существует тесная связь Для каждого алгоритма существует исчисление, порождающее область определения этого алгоритма, более того, можно указать исчисление, порождающее те и только те пары $(x,y)$, для которых $\Psi(x) = y$. С другой стороны, для каждого исчисления существует алгоритм, область определения которого совпадает с множеством, порождаемым исходным исчислением. Каждый алгоритм задает функцию на своей области применимости, ее значения равны результату алгоритма $\Psi(x)$. Применительно к исчислению можно говорить о породимых, разрешимых и перечислимых множествах. Тезисы Черча и Поста формулируются, соответственно, для вычислимой модели алгоритма и порождающей модели исчисления. С помощью общего понятия исчисления можно глубже осмыслить многие фундаментальные понятия математической логики. В частности, знаменитую теорему Геделя о полноте, утверждающую, что все истинные формулы логики предикатов 1-го порядка могут быть порождены некоторым исчислением. Другая знаменитая теорема Геделя о неполноте утверждает, что множество всех истинных формул арифметики (а, значит, и множество всех общезначимых формул логики предикатов 2-го порядка) не может быть порождено никаким исчислением. На базе понятия исчисления можно изложить всю дескриптивную теорию алгоритмов (наличие или отсутствие алгоритма без оценки затрат на достижение этой цели). Вычислимая функция – это функция, вычислимая каким-либо алгоритмом: при применении к какому-нибудь входу вычисляющий алгоритм должен не только давать результат, совпадающий со значением функции на этом входе, если такое значение существует, но и не давать никакого результата, если функция не определена на данном входе. Породимое множество – это множество, порождаемое каким-либо исчислением. Перечислимое множество – это либо множество значений всюду определенной вычислимой функции натурального аргумента, либо пустое множество. Обе теоремы Геделя можно сформулировать в терминах перечислимости и неперечи-слимости соответствующих множеств. Множество называется разрешимым, или распознаваемым, если оно содержится в некотором породимом множестве $X$ и для него существует разрешающий алгоритм. Алгоритм $\Psi$ называется разрешающим алгоритмом для подмножества $A$ множества $X$, если множество допустимых входов для $\Psi$ совпадает с X и $\Psi$ отвечает на все вопросы типа “$x \in X$ & $x \in A$”.Проблема отыскания такого алгоритма называется проблемой разрешения для множества $A$. Существование неразрешимого множества $A$ в породимом множестве $X$ означает, что никаким алгоритмом нельзя определить, будет ли произвольный элемент из $A$ когда-либо порожден. В качестве примера неразрешимого множества приведем «проблему остановки машины Тьюринга» [1]. Машину Тьюринга и ее программу можно представить словом в некотором алфавите и подать на вход другой машине Тьюринга или ей самой. «По коду машины Тьюринга и входу, записанному на ленте, надо за конечное время определить, заканчивает машина Тьюринга работу на этом слове или работает бесконечно». Это и есть проблема остановки (останова) машины Тьюринга. Предположим, что машина Тьюринга, решающая задачу останова существует, обозначим ее $H(m, s)$. Эта машина анализирует код машины m и содержимое ее ленты $s$. Если $m$ допускает $s$, то $H$ допускает, в противном случае $H$ останавливается, не допуская (обозначим это состояние через $q_{no}$). Таким образом, машина $H(m, s)$ всегда останавливается, в отличие от машины $m$ на слове $s$. Построим машину $T$ ($str$), выполняющую следующий алгоритм: имитируется работа машины $H(str, str)$ и, если H перешла в $q_{no}$ , то $T$ ($str$) допускает, в противном случае $T$ ($str$) работает бесконечно. Таким образом, $T$ ($str$) допускает, если $H(str, str)$ работает бесконечно, в противном случае $T$ ($str$) работает бесконечно. Посмотрим, как работает машина $T$ на своем коде $t$: если $T$ «зависает» на $t$ (работает бесконечно или переходит в состояние $q_{no}$), то $H$(t, t) допускает, т.е. машина $T$ допускает на своем коде $t$; если $T$ на $t$ допускает, то $H(t, t)$ переходит в состояние $q_{no}$, т.е. $T$ на коде $t$ не допускает. Получено противоречие. Следовательно, машина, разрешающая проблему остановки, не существует. Изучая неразрешимые породимые множества, возни-кающие в математической практике, обнаружили, что неразрешимые проблемы сводятся друг к другу. Это позволило построить метод доказательства неразрешимости проблемы, сводя к ней известную неразрешимую проблему.

Дальше речь пойдет о разрешимых проблемах и машино-независимой теории сложности, выработке сложностных оценок для алгоритмов и вычислений. Первый путь в этом направлении - поиск сложностных функций, независимых от выбора вычислительной (или порождающей модели). При оценке сложности применяются методы редукции и сведения к другим задачам с известными оценками сложности

$f(n) \le c\, f(n/a) + b\, g(n) + dn$, где $a, b, c, d$ – некоторые положительные константы.

Свойства функции сложности:

$$a)\ f(n) \ge n; \qquad b)\ k\, f(n) \le f(kn),\ k > 1.$$

Лемма 1. Если функции $f(n)$ *и* $g(n)$ удовлетворяют свойствам a), b) и при $a > 1$, $b > 0$, $c > 0$ выполняется основное неравенство, то $f(n) = O(g(n))$.

Лемма 2. Если при некоторых константах $a > 0$, $c > 0$, $d > 0$

$$f(1) = d,$$
$$f(n) = c\, f(n/a) + dn,$$

то при $n > 1$ $f(n) = O(n)$ при $a > c$; $f(n) = O(n \log n)$ при $a = c$; $f(n) = O(n^{\log_a c})$ при $a < c$.

*Пример* использования леммы 2 – умножение методом Карацубы: $A$, $B$ –два $n$–разрядных числа. Разобьем их на два слагаемых $A = 2^k A_1 + A_0$, $B = 2^k B_1 + B_0$. Тогда

$$AB = (2^{2k} + 2^k)\, A_1 B_1 + 2^k (A_0 - A_1)(B_0 - B_1) + (2^k + 1) A_0 B_0.$$

Таким образом, для вычисления произведения двух $n$-разрядных чисел надо выполнить три умножения $n/2$-разрядных чисел и некоторое количество сложений, вычитаний и сдвигов. получим оценку $f(n) = 3 f(n/2) + cn$, $c > 0$. По лемме 2: сложность умножения $M(n) < f(n) = O(n^{\log_2 3})$.

Второй путь – это попытка включения параметров вычислительной модели в сложностную функцию (например, число лент машины Тьюринга и мощность алфавита). Этот подход рассматривается в рамках теории сложности вычислений.

Третий путь – аксиоматический, предложенный Блюмом. Рассматриваются две функции: операторная $F$: $E \times X \to Y$ и мера сложности $C$: $E \times X \to N$, удовлетворяющие аксиомам:

1) $F(i, x)$ определено тогда и только тогда, когда $C(i, x)$ определено;
2) множество $\{(i, x, y): C(i, x) = y\}$ - разрешимо.

Примерами мер сложности, удовлетворяющих аксиомам Блюма, являются время и емкость вычислений на машинах Тьюринга. Блюмом были доказаны две теоремы о мерах сложности.

*Теорема о рекурсивной связи различных мер сложности.* Пусть $C_1(i, x)$, $C_2(i, x)$ – две меры сложности для одной и той же операторной функции. Тогда существует вычислимая функция $G$: $X$ x $N \rightarrow N$, такая, что для всех $i$, $C_2(i, x) \leq G(x, C_1(i, x))$ выполняется для всех $x$, для которых $F(i, x)$ определена всюду, кроме конечного числа.

*Теорема об ускорении*. Пусть $C$ – мера сложности, $f$ – всюду *определенная* вычислимая функция из $N$ в $N$. Тогда существует разрешимое подмножество $A$ множества $X$, что для любого $i \in E$, для которого $F\ (i, x) = \chi_A\,(x)$, для всех $x$, существует такое $j$, что $F(i, x) = \chi_A\,(x)$, для всех $x$ кроме конечного числа, верно неравенство $C\,(i, x)\ \geq\ f\,(C\,(j, x))$.

«*Неразрешимость» в теории сложности как невозможность решить проблему на компьютере.* Практически разрешимые проблемы входят в сложностной класс $P$, полиномиально разрешимых задач, а имеющие более чем полиномиальную сложность, принадлежат классу $NP$. В классе $NP$, в свою очередь, выделяется подкласс $NP$ – полны проблем, называемых труднорешаемыми. Для доказательства принадлежности проблемы к классу $NP$ – полных задач используется известный в логике метод сведения известной труднорешаемой проблемы $K_1$ к проблеме $K_2$, последняя не может быть легче первой. Однако методы сведения, используемые в логике и в теории сложности различаются как принципами, на которых они базируются, так и ограничениями на сводящий алгоритм. *Доказательство неразрешимости проблем* основано на общих математических принципах: понятии алгоритма и формальной системы. Утверждение о *труднорешаемости проблемы* базируется на недоказанном принципе, что $P \neq NP$ и факте, что для всех труднорешаемых проблем не известно ни одной детерминированной машины Тьюринга, решающей эту проблему за полиномиальное время. Что касается сводящего алгоритма, то для доказательства труднорешаемости задачи уже недостаточно просто наличия сводящего алгоритма, но сам сводящий алгоритм должен уже иметь полиномиальную сложность, чтобы не переносить свою труднорешаемость на проблему, к которой с его помощью сводится известная труднорешаемая проблема.

Рассмотрим класс проблем, включающий классы $P$ и $NP$, этот класс определяется машинами Тьюринга с полиномиальной емкостной сложностью $PS$ (= $NPS$). В полиномиальном пространстве существуют полные проблемы, к которым полиномиально (по времени) сводятся все проблемы данного класса. Если примером $NP$-полной проблемы служит класс выполнимых булевых функций, то примером $PS$-полной проблемы – класс истинных булевых функций с кванторами.

Труднорешаемые задачи, в первую очередь $NP$-полные, не имеющие полиномиального алгоритма для нахождения оптимальных решений, можно "решать" с помощью приближенных алгоритмов, решение которых не оптимальное, а некоторое приближение к нему, приемлемое на практике.

**Библиографический список**

1. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений. – Москва, Санкт-Петербург, Киев: Изд. дом "Вильямс". – 2002. – 528 с.

**УДК 579.64**

# Автомодельное решение задачи о движении газогидрата при изменении температуры

***П.В. Гилев, А.А. Папин***
*АлтГУ, г. Барнаул*

**Постановка задачи.** В настоящее время залежи природных газовых гидратов рассматриваются как потенциальный источник природного газа. Приоритетной является проблема развития технологий его извлечения [1]. Математические модели процессов, связанные с разработкой газогидратов, основаны на моделях тепловой многофазовой фильтрации в деформируемых пористых средах с учетом фазовых переходов и свободных границ.

В настоящей работе рассматриваются вопросы обоснования модельной задачи движения жидкости в деформируемой пористой среде. Изучается следующая система дифференциальных уравнений:

$$\frac{\partial\rho}{\partial t} + \vec{v}\cdot\nabla\rho + \rho\nabla\cdot\vec{v} = 0, \frac{\partial\vec{v}}{\partial t} + (\vec{v}\cdot\nabla)\vec{v} = \rho^{-1} div\Sigma,$$
$$\frac{\partial\vec{u}}{\partial t} = \left(I - \frac{\partial\vec{u}}{\partial\vec{x}}\right)\vec{v}, m \cong m_0\left(1 - \frac{1}{2}\nabla\cdot\vec{u}\right), \qquad (1)$$