Алгоритмически неразрешимые проблемы

Одним из важнейших свойств алгоритма является массовость, то есть способность решать не одну конкретную задачу, а целый класс однотипных задач.

Как показали исследования, существуют такие классы задач, для которых не существует единого универсального приема решения. Такие задачи в теории алгоритмов получили название алгоритмически неразрешимых проблем. Заметим, что алгоритмическая неразрешимость не означает, что такие задачи нельзя решить. Речь идет о невозможности решения одним и тем же приемом.

Примером алгоритмически разрешимой проблемой проблемы является доказательство тождеств в алгебре. Известна последовательность действий: раскрыть все скобки, привести подобные члены и перенести их все на одну сторону, если $0 = 0$, то тождество истинно, иначе ложно.

Вместе с тем, уже ранние исследования в области построения алгоритмов показали неразрешимость некоторых задач. Одним из первых результатов подобного рода является доказательство неразрешимости проблемы распознавания выводимости в математической логике, известное в литературе под названием «теорема Черча» (А. Черч, 1936 г.). Основная идея состоит в том, что Черч смог показать нерекурсивность функции, решающей данную задачу, в результате чего стало ясно, что искать алгоритм бессмысленно.

Одним из примеров алгоритмически неразрешимой проблемы является проблема распознавания «самоприменимости». Существуют самоприменимые и несамоприменимые алгоритмы. Например, тождественный алгоритм в любом алфавите А, содержащем две и более букв, является самоприменимым, способным перерабатывать любое входное слово $p \in$ А в само себя.

С другой стороны, нулевой алгоритм в алфавите А является несамоприменимым. Этот алгоритм задается схемой, содержащей единственную подстановку «$\rightarrow y$»($y \in A$). По определению алгоритм неприменим ни к какому входному слову, а значит и к своему изображению.

Проблема самоприменимости алгоритмов состоит в том, чтобы установить за конечное количество шагов по схеме любого алгоритма является ли он самоприменимым или нет. Рассмотрим проблему детальнее посредством алгоритмической системы машины Тьюринга.

Пусть на ленте МТ зафиксирована некоторая конфигурация. Тогда возможны два случая:

1. МТ применима к этой конфигурации, тогда она после N шагов остановится в заключительном состоянии.

2. МТ неприменима к этой конфигурации, то есть она никогда не достигнет заключительного состояния и будет продолжать свою работу бесконечно.

Теорема. *Проблема распознавания самоприменимости алгоритма является неразрешимой.*

Рассмотрим вариант универсальной МТ, когда на ленте изображен ее собственный шифр (таблицы соответствия и начальная конфигурация) в алфавите машины. Естественно считать, что если МТ обработает этот шифр — она самоприменима, иначе — несамоприменима.

Предположим, что такая МТ, анализирующая собственный шифр и выносящая решение о самоприменимости, существует. Тогда в этой МТ всякий самоприменимый шифр перерабатывается в некоторый символ ν (положительный ответ), а всякий несамоприменимый шифр — в символ τ (отрицательный ответ).

При таком допущении можно построить такую МТ', которая перерабатывает несамоприменимые шифры в τ (как и МТ), но неприменима к самоприменимым шифрам. Это достигается введением такого изменения таблицы соответствия, когда после появления символа ν вместо остановки МТ' бесконечно повторяла бы $\nu, \nu, \nu, ...$

Итак, МТ' применима ко всякому несамоприменимому шифру и неприменима к самоприменимым шифрам. Это приводит к противоречию.

1. Пусть МТ' самоприменима, то есть она она применима к собственному шифру МТ' и останавливается, но тогда она несамоприменима, так как вырабатывает τ.

2. Пусть МТ' несамоприменима, что означает ее применимость к ее собственному шифру МТ', то есть она самоприменима.

Указанное противоречие и есть доказательство теоремы, так как оно опровергает утверждение о существовании МТ.

Из этой теоремы вытекает алгоритмическая неразрешимость более широкой проблемы — проблемы распознавания применимости, заключающейся в необходимости установить применимость любой МТ к любой заданной конфигурации.

В терминах нормальных алгоритмов проблема самоприменимости может быть сформулирована следующим образом: «Требуется найти единственный прием, позволяющий за конечное число шагов по схеме любого алгоритма A узнать, является ли A самоприменимым или нет».

С точки зрения НА единый конструктивный прием — это некий нормальный алгоритм B, определенный на любом слове p, которое является изображением произвольного НА A, переводящий это слово в два различных фиксированных слова q_1 и q_2 в зависимости от того, будет ли алгоритм A самоприменимым или нет.

На любом входном слове L, не являющемся изображением какого-либо НА, алгоритм B также должен быть определен. Иначе, после некоторого числа шагов работы, было бы неизвестно, изображением какого алгоритма (самоприменимого или нет) является слово L. Результат обработки алгоритмом B слова L, не являющегося изображением алгоритма, должен отличаться от q_1 и q_2.

Предположим, что B существует. В таком случае существует и НА C в том же алфавите X, что и алгоритм B, определенный на всех тех и только тех словах в X, которые являются изображениями несамоприменимых алгоритмов.

Для этого построим НА D в алфавите X, область определения которого состоит из одного слова q_2. Его можно задать, например, в виде суперпозиции двух НА $D1$ и $D2$. $D1$ задается схемой, состоящей из одной подстановки «$q_2 \to *$», а $D2$ — схемой из подстановок «$x_i \to x_i$», где x_i принимает последовательно значение всех букв алфавита X. $D1$ переводит в пустое слово только слово q_2, а область определения $D2$ — пустое слово. Область определения суперпозиции D алгоритмов $D1$ и $D2$ содержит лишь q_2. Построив алгоритм D, образуя суперпозицию с ним алгоритма B и нормализуя эту суперпозицию, получим НА C в алфавите X, область определения которого состоит из тех и только тех слов X, которые являются записями несамоприменимых алгоритмов. Однако подобное свойство алгоритма C внутренне противоречиво, поскольку к своему собственному изображению алгоритм C не может быть одновременно применимым и неприменимым.

Действительно, если алгоритм C применим к своему изображению, то он самоприменим. Это противоречит тому, что алгоритм C, в силу своего построения, должен быть применим только к несамоприменимым алгоритмам. С другой стороны, если алгоритм C неприменим к своему изображению, то он является несамоприменимым. Но тогда он должен быть применим к своему изображению, так как он применим к изображениям всех несамоприменимых алгоритмов. Следовательно, алгоритм C — самоприменим. Таким образом, найдено логическое противоречие, демонстрирующее алгоритмическую неразрешимость всей проблемы (при условии, что принцип нормализации справедлив).

Природа противоречия, использованного в ходе рассуждений, имеет более глубокие корни и связана с парадоксом Рассела («множество всех множеств, не содержащее себя в качестве своего элемента»).

Этот же метод «от противного» может быть использован при доказательстве неразрешимости проблемы эквивалентности слов для ассоциативного исчисления.

Определение. *Ассоциативным исчислением называется совокупность всех слов в некотором алфавите A с конечной системой подстановок.*

Чтобы задать ассоциативное исчисление достаточно указать алфавит A и систему подстановок. Подстановки бывают вида «$p \leftrightarrow q$» или «$p \to q$», где p, q — слова в алфавите A. Неориентированная подстановка позволяет замену в прямом и обратном направлениях, ориентированная — лишь в прямом.

Допустим, задан алфавит $A = \{a, b, c\}$ и одна подстановка «$ab \leftrightarrow bcb$», а также $p = abcbcbab$. Заменим вхождения ab: $bcbcbcbbcb$, а затем используем подстановку в обратном направлении: $abcbcbab \to aabcbab \to aaabab$.

Если слово r может быть получено из слова s применением лишь одной подстановки, то говорят, что r и s смежные слова. Если слово r может быть получено из слова s применением конечного числа подстановок, то говорят, что смежные слова образуют «дедуктивную цепочку от s к r». Наконец, слова s и r называются «эквивалентными» ($s \sim r$), если существует дедуктивная цепочка от s к r.

Для отношения эквивалентности слов справедливо следующее.

1. Рефлексивность: $r \sim r$.

2. Симметричность: $r \sim s \leftrightarrow s \sim r$.

3. Транзитивность: $r \sim t, t \sim s \rightarrow r \sim s$.

Проблема эквивалентности в ассоциативном исчислении состоит в необходимости определения для для любых двух слов в данном ассоциативном исчислении эквивалентны они или нет.

Впервые, в 1914 г., данная проблема была сформулирована А. Туэ. Он предложил некоторые алгоритмы распознавания эквивалентности слов для частных специальных исчислений с повторениями сочетаний букв в словах заданного алфавита Σ. Туэ назвал слово «бесквадратным», если оно не содержит сочетаний вида $x^2 = xx$ (или «бескубным», если оно не содержит сочетаний вида x^3), где $x \neq \lambda$.

Суть решаемой Туэ проблемы состояла в построении бесквадратных (бескубных) слов максимальной длины в Σ. После этого многие математики предпринимали усилия для построения такого общего алгоритма, который бы для любого ассоциативного исчисления и для любой пары слов в нем позволял бы установить, являются ли они эквивалентными. В 1947 г. Э. Пост и в 1964 г. А.А. Марков независимо друг от друга построили примеры ассоциативных исчислений, где проблема эквивалентности слов оказалась алгоритмически неразрешимой. Это позволило сделать вывод о том, что искомого алгоритма не существует в принципе. В 1955 г. П.С. Новиков доказал алгоритмическую неразрешимость проблемы распознавания эквивалентности слов для ассоциативного исчисления специального вида — теории групп.

Следует отметить, что примеры, построенные для опровержения алгоритмической разрешимости проблемы эквивалентности слов в ассоциативных исчислениях, оказались очень сложными и громоздкими, включающими в себя сотни допустимых подстановок. Однако в 1959 г. Г. Цейкинсну удалось построить пример ассоциативного исчисления, имеющего всего семь допустимых подстановок, для которого проблема распознавания эквивалентности слов оказалась также алгоритмически неразрешимой.

Пусть задан алфавит $A = \{a, b, c, d, e\}$ и следующая система подстановок.

- $ac \rightarrow ca$;
- $ad \rightarrow da$;

- $bc \to cb$;
- $eca \to ae$;
- $bd \to db$;
- $abac \to abace$;
- $be \to edb$.

Рассмотрим некоторое слово $p_1 = abcde$. Используя подстановки, построим дедуктивную цепочку $acbde \to cabde \to cadbe \to cadedb$. Соответственно, $abcde \sim cadedb$. Рассмотрим другое слово $p_2 = aaabb$. Очевидно, к нему неприменима ни одна из данных подстановок, а это значит, что p_2 не имеет ни смежных, ни, следовательно, эквивалентных слов. Цейкинс успешно доказал, что для любой пары слов в этом ассоциативном исчислении нет и не может быть единого алгоритма распознавания эквивалентности.

Существует не один десяток примеров алгоритмически неразрешимых проблем. Одним из важнейших является теорема Геделя о неполноте арифметической логики (совокупности аксиом и правил вывода в элементарной теории чисел).

Определение. *Логика называется полной, если в рамках ее можно доказать истинность или ложность каждого утверждения.*

Определение. *Логика называется непротиворечивой, если она свободна от противоречий (например, в ней нельзя получить одновременно истинное утверждение* A *и ложное* ¬A*).*

Теорема. *Каждая адекватная ω-непротиворечивая арифметическая логика неполна.*

Математический смысл теоремы в том, что в арифметической логике существуют истинные утверждения о целых положительных числах, которые нельзя вывести и доказать средствами этой логики. Кроме того, Гедель показал, что невозможно доказать непротиворечивость арифметической логики теми методами, которые выразимы в данной логике.

Результаты работы Геделя являются фундаментальными и одними из наиболее значимых для современной математики

Первой известной аксиоматической системой принято считать геометрию Евклида(3 век до нашей эры). В ее основе лежит набор определений и аксиом, которые отражают простейшие геометрические свойства, не вызывающие сомнений и подтвержденные всем опытом человечества. В их

число входит и аксиома о параллельных прямых: «Если две прямые, лежащие в одной плоскости, при пересечении их какой-нибудь третьей образуют внутренние углы, сумма которых меньше двух прямых, то эти прямые пересекаются по ту сторону от третьей, на которой сумма указанных углов меньше двух прямых».

Данная аксиома является наиболее сложной, ее пытались вывести из какой-либо другой, но безуспешно. В истории математики она получила название «темное пятно Евклида». Не удалось также доказать полноту и противоречивость системы аксиом Евклида Декарту, Лейбницу, Гауссу и многим другим выдающимся математикам.

В 1826 г. бессмысленность этих попыток впервые строго доказал русский математик Н.И. Лобачевский. В частности он показал, что данная аксиома Евклида невыводима из остальных аксиом. Совокупность его научных трудов получила название «геометрия Лобачевского» («неевклидова геометрия»). Одним из результатов Лобачевского является отрицание истинности одной из аксиом Евклида: «Если даны прямая и точка, не лежащая на ней, то существует только одна прямая, которая проходит через данную точку параллельно первой прямой». Таким образом, в геометрии Лобачевского нет параллельных прямых, а, в то же время, геометрия Евклида есть частный случай геометрии Лобачевского. Немецкий математик Б. Риман доказал, что неевклидова геометрия может быть непротиворечивой только в том случае, если удастся доказать непротиворечивость геометрии Евклида. Другой выдающийся немецкий математик Д. Гильберт в свою очередь показал, что геометрия Евклида могла бы быть непротиворечивой, если бы удалось доказать непротиворечивость арифметики. Для этого нужно было найти вариант такой арифметической логики, которая оказалась бы полной, то есть в которой можно вывести в качестве теорем все истинные утверждения о целых положительных числах.

Теперь вернемся к теореме Геделя. Оказалось, что такой арифметической логики нет и быть не может, а это значит, что существует бесконечное число проблем элементарной теории чисел, решение которых невозможно никаким аксиоматическим методом.

Одним из практических следствий теоремы Геделя является то, что алгебра Буля в алфавите $A = \{0, 1\}$, который есть подмножество $B = \{0, 1, 2, ..., 9\}$ или $A \subseteq B$, может иметь также истинные выражения и формулы, которые невозможно вывести из системы тождественных соотноше-

ний. Этот результат напрямую касается вопросов проектирования систем обработки информации.