

Лекция 17. Примеры неразрешимых задач

Дискретная математика, ВШЭ, факультет компьютерных наук

(Осень 2014 – весна 2015)

Мы ввели машины Тьюринга для того, чтобы доказывать алгоритмическую неразрешимость конкретных задач. Приведём один из простейших примеров: задачу достижимости на графе подстановок слов.

Как вам известно, задача достижимости (можно ли из одной вершины попасть в другую, двигаясь по рёбрам) на конечном ориентированном графе разрешима. Более того, есть весьма эффективные алгоритмы её решения.

Подходящее обобщение этой задачи на бесконечные графы даёт уже алгоритмическую неразрешимость. Начнём с описания такого обобщения.

1 Формулировка задачи

Здесь мы будем рассматривать бесконечные ориентированные графы, причём такие, что выходная и входная степень каждой вершины конечна.

Нужно договориться о способе задания графа и его вершин: для бесконечного графа это можно делать существенно различными способами.

Мы выберем такой способ задания (ор)графа. Множество вершин — это множество слов в некотором алфавите Σ . А рёбра задаются *правилами подстановки*. Каждое правило имеет вид

$$L \rightarrow R,$$

где L, R — слова в алфавите Σ . Из слова x ведёт ребро в слово y по правилу подстановки $L \rightarrow R$, если $x = uLv$, $y = uRv$.

Пример 1. Рассмотрим слова (последовательности букв) в русском алфавите. Тогда из слова *входная* ведёт ребро в слово *выходная* по правилу подстановки *ход* $\rightarrow$ *ыход*.

Таким образом, одно правило подстановки даёт бесконечное количество рёбер. Мы разрешим использовать несколько правил подстановки:

$$L_1 \rightarrow R_1,$$

$$L_2 \rightarrow R_2,$$

$$\dots$$

$$L_n \rightarrow R_n,$$

и множество рёбер графа будет состоять из всех рёбер, отвечающих какому-то из этих правил.

Будем также использовать обозначение $u \xrightarrow{\mathcal{R}} v$ для пары вершин нашего графа (т.е. слов), связанных одним из правил подстановки из множества правил подстановки $\mathcal{R} = \{L_i \rightarrow R_i\}$. Если из слова u можно попасть в слово v , двигаясь по рёбрам орграфа, будем указывать это как $u \xrightarrow{*} v$ (другими словами, это обозначение для отношения достижимости).

Задача достижимости. Задан граф на множестве слов в алфавите Σ набором правил подстановки $\mathcal{R} = \{L_i \rightarrow R_i\}$ и два слова $u, v \in \Sigma^*$. Верно ли, что $u \xrightarrow{*} v$?

В некоторых случаях решить задачу достижимости легко.

Пример 2. Пусть есть ровно одно правило подстановки в алфавите $\{a, b\}$:

$$ab \rightarrow ba.$$

Тогда верно, что $aaaabbaab \xrightarrow{*} baabaabaa$. (Количество букв a и b одинаково, а правило подстановки позволяет поменять местами любую пару различных букв.)

Пример 3. Пусть правила замены слов в алфавите $\{a, b\}$ имеют вид:

$$aba \rightarrow baabbb,$$

$$bbba \rightarrow a,$$

$$abab \rightarrow bbbb.$$

Тогда неверно, что $ababbbaa \xrightarrow{*} bbbaaba$. (Подстановки не меняют чётности количества букв a в слове.)

Однако общего «рецепта» решения задачи достижимости, т.е. алгоритма, нет.

Чтобы говорить о существовании или несуществовании алгоритма решения задачи достижимости, нужно представить условия задачи словом в конечном алфавите. Будем использовать те же правила, что и при описании машин Тьюринга. Каждый символ алфавита $i \in \Sigma = \{0, 1, \dots, n\}$ кодируется двоичным словом

$$\langle i \rangle = \underbrace{00 \dots 0}_i \underbrace{100 \dots 00000}_{n-i},$$

слово $u = u_1 \dots u_k$, $u_i \in \Sigma$, кодируется словом

$$\langle u \rangle = \# \langle u_1 \rangle \# \langle u_2 \rangle \# \dots \# \langle u_k \rangle \#$$

в алфавите $\{0, 1, \#\}$. Правило подстановки $L \rightarrow R$ записывается как коды слов L и R , разделённые символом $\rightarrow$ и окружённые разделителями $\triangleleft, \triangleright$:

$$\triangleleft \langle L \rangle \rightarrow \langle R \rangle \triangleright.$$

Код набора правил $\mathcal{R} = \{L_i \rightarrow R_i\}$ — это просто запись всех правил подстановки одного за другим.

2 Неразрешимость задачи достижимости для графа подстановок слов

Теорема 1. Не существует алгоритма, который по слову $\langle \mathcal{R} \rangle \odot \langle u \rangle \odot \langle v \rangle$, где $\mathcal{R}$ — набор правил подстановки, u, v — слова, выдавал бы результат 1, если $u \xrightarrow[\mathcal{R}]{} v$, и 0 в противном случае.

Доказательство. Используем неразрешимость проблемы остановки машины Тьюринга. Общий план доказательства такой: предположим, что существует алгоритм для решения задачи достижимости. Тогда мы построим алгоритм решения задачи остановки МТ следующего вида: алгоритм преобразует вход $\langle M \rangle \odot \langle x \rangle$ в описание набора правил $\mathcal{R}$ и двух слов u, v ; после этого вызывает алгоритм решения задачи достижимости и выдаёт в качестве результата результат работы этого алгоритма.

Чтобы описанный алгоритм был корректным, нужно выполнение такого условия: машина M останавливается на входе x тогда и только тогда, когда $u \xrightarrow[\mathcal{R}]{} v$.

Мы сейчас опишем такое преобразование машины и её входа в набор правил подстановки и два слова, для которого это свойство выполняется.

Итак, пусть имеется МТ $M = (A, Q, \delta, q_0, \Lambda)$ и её входное слово x . Заметим, что за один такт работы конфигурация МТ изменяется незначительно — изменения затрагивают лишь небольшую окрестность символа из множества состояний Q , который обязательно присутствует в конфигурации.

Мы хотим описать это изменение как результат применения правил подстановки. При этом удобно применять правила подстановки к словам чуть более общего вида, чем конфигурации.

В качестве алфавита для графа подстановок слов возьмём множество

$$\Sigma = A \cup Q \cup \{\Lambda_0, f\}, \quad \{\Lambda_0, f\} \cap (A \cup Q) = \emptyset.$$

Дополнительный символ Λ_0 будет использоваться для расширенного описания конфигураций. А именно, конфигурация uqv машины M будет представляться любым словом вида $\Lambda_0 \Lambda^k uqv \Lambda^s \Lambda_0$. Другими словами, мы разрешаем иметь слева и справа от символов конфигурации произвольное количество пустых символов, окаймлённое дополнительным символом Λ_0 .

Дополнительный символ f будет играть вспомогательную роль, которая станет ясна позднее.

Достижимость будет проверяться для пары слов $u = \Lambda_0 q_0 x \Lambda_0$ и $v = \Lambda_0 f \Lambda_0$.

Осталось описать набор правил подстановки Δ . Правила будут двух типов.

Правила первого типа описывают такт работы МТ. Чтобы их задать, нужно аккуратно рассмотреть все возможные случаи положения и движения головки на ленте и выписать правила преобразования конфигурации.

Для удобства описания расширим таблицу переходов на множество $(A \cup \{\Lambda_0\}) \times Q$ по правилу $\delta(\Lambda_0, q) = \delta(\Lambda, q)$ для любого $q \in Q$, это формальная запись такого свойства «с точки зрения таблицы переходов символ Λ_0 не отличается от пустого символа Λ ».

Пусть $\delta(a, q) = (b, q', 0)$. По такой строчке таблицы переходов добавим правила

$$qa \xrightarrow{\Delta} q'b, \quad a \neq \Lambda_0, \quad (1)$$

$$q\Lambda_0 \xrightarrow{\Delta} q'b\Lambda_0. \quad (2)$$

Если $\delta(a, q) = (b, q', 1)$, то добавим правила

$$qa \xrightarrow{\Delta} bq', \quad a \neq \Lambda_0, \quad (3)$$

$$q\Lambda_0 \xrightarrow{\Delta} bq'\Lambda_0. \quad (4)$$

Наконец, если $\delta(a, q) = (b, q', -1)$, то добавим правила

$$xqa \xrightarrow{\Delta} q'xb, \quad \text{для всех } x \in A, a \neq \Lambda_0, \quad (5)$$

$$\Lambda_0 qa \xrightarrow{\Delta} \Lambda_0 q' \Lambda b, \quad a \neq \Lambda_0, \quad (6)$$

$$xq\Lambda_0 \xrightarrow{\Delta} q'xb\Lambda_0, \quad \text{для всех } x \in A, \quad (7)$$

$$\Lambda_0 q\Lambda_0 \xrightarrow{\Delta} \Lambda_0 q' \Lambda b \Lambda_0. \quad (8)$$

Правила второго типа задают преобразования слов после окончания работы МТ.

Для каждой пары (a, q) , $a \in A \cup \{\Lambda_0\}$, на которой таблица переходов не определена, добавим правило

$$qa \xrightarrow{\Delta} fa. \quad (9)$$

Добавим также правила

$$af \xrightarrow{\Delta} f, \quad \text{для всех } a \in A, \quad (10)$$

$$fa \xrightarrow{\Delta} f, \quad \text{для всех } a \in A. \quad (11)$$

На этом построение набора правил закончено. Проверим, что построенные Δ , u , v удовлетворяют сформулированному выше требованию: машина M останавливается на x тогда и только тогда, когда $\Lambda_0 q_0 x \Lambda_0 \xrightarrow[\Delta]{*} \Lambda_0 f \Lambda_0$.

Прежде всего заметим, что к слову вида

$$\Lambda_0 \Lambda^k w \Lambda^s \Lambda_0,$$

где w — конфигурация МТ, применимо не более одного правила первого типа, и его применение даёт слово такого же вида

$$\Lambda_0 \Lambda^{k'} w' \Lambda^{s'} \Lambda_0,$$

где w' — конфигурация после такта работы машины M на конфигурации w . (Для каждого правила (1–8) это утверждение проверяется непосредственным применением правила выполнения такта работы МТ и определения конфигурации.)

Значит, если МТ M не останавливается на входе x , то преобразованиями по данной системе правил будут получаться слова, в которые обязательно входит символ из Q , т.е. слово $\Lambda_0 f \Lambda_0$ получить из начального слова невозможно.

Пусть МТ M останавливается на входе x . Заметим, что возможна ровно одна подстановка для слова вида $\Lambda_0 \Lambda^k w \Lambda^s \Lambda_0$, где w — конфигурация МТ. Поэтому возможно выполнение подстановок до тех пор, пока не будет достигнута финальная конфигурация. В финальной конфигурации применимо правило (9). После этого применением правил (10, 11) возможно убрать из слова все символы алфавита A . Останется в точности искомое слово $\Lambda_0 f \Lambda_0$.

Описание правил подстановки было вполне конструктивным. Поэтому легко поверить, что существует алгоритм, который по описанию МТ M и входа x строит описание Δ и слов u , v . Если говорить о (многоленточной) МТ, решающей такую задачу, нужно разбить её построение на несколько шагов:

- Определение размера алфавита Σ и построение МТ, которая преобразует код символа в описании M в код соответствующего символа в алфавите Σ .
- Преобразование описания x в описание слов $u = \Lambda_0 q_0 x \Lambda_0$ и $v = \Lambda_0 f \Lambda_0$.
- Запись описания правил первого типа по каждой строчке таблицы переходов.
- Перечисление всех пар (a, q) , для которых таблица переходов не определена и запись правил второго типа, отвечающих таким парам.
- Запись остальных правил второго типа, для этого нужно перечислить все символы алфавита A .

□

Описанное преобразование МТ и её входа в набор правил подстановки обладает дополнительными интересными свойствами. Скажем, легко заметить, что правила зависят только от машины, а не от входа. Вспомнив о существовании универсальной машины, получаем такое следствие.

Следствие 1. *Существует такой граф подстановок, что задача достижимости для этого графа алгоритмически неразрешима.*

Неориентированные графы также можно задавать правилами подстановки. Но в этом случае подстановки двусторонние: можно не только заменять левую часть правила на правую, но и наоборот: правую на левую.

Задача 1. Докажите, что задача достижимости для неориентированных графов, заданных правилами подстановки, алгоритмически неразрешима.

Подсказка: используйте то же самое преобразование, что и в доказательстве теоремы, и докажите, что даже при двусторонних заменах слово v достижимо из слова u тогда и только тогда, когда машина M останавливается на входе x .

3 Другие примеры

Доказательства неразрешимости следуют обычно тому же плану, который мы реализовали в доказательстве неразрешимости задачи достижимости.

Алгоритмы того вида, который использован в этом доказательстве, получили специальное название — *сводимости*. Легко понять, что сводимости транзитивны: скажем, для доказательства неразрешимости какой-нибудь задачи можно теперь использовать неразрешимости задачи достижимости и т.д.

Известно довольно много неразрешимых задач и далеко не всегда по формулировке задачи легко догадаться, что в терминах этой задачи можно выразить вопрос об остановке машины Тьюринга. Приведём без доказательства несколько ярких примеров неразрешимых задач.

Пример 4 («Бесконечный пазл»). Задан конечный набор единичных квадратов, стороны которых покрашены в некоторые цвета (конечное число). Спрашивается, можно ли замостить всю координатную плоскость квадратами заданных типов так, чтобы все квадраты соседствовали по сторонам одного цвета?

Замечание 1. Более простой вариант этой задачи получается, если дополнительно потребовать, чтобы замощение содержало квадрат некоторого заданного типа. В таком случае доказательство неразрешимости лишь чуть сложнее описанного выше доказательства неразрешимости проблемы достижимости. Нужно сообразить, как задать последовательность конфигураций МТ в виде строк плоскости (для этого потребуются ещё вспомогательные строки между строками, отвечающими конфигурациям).

Попробуйте придумать такое доказательство!

Пример 5 («Задача вымирания»). Задан конечный набор матриц одного порядка с целыми коэффициентами. Спрашивается, можно ли выразить нулевую матрицу как произведение матриц из указанного набора (матрицы в произведение могут входить в любом порядке).

Пример 6 («Диофантовы уравнения»). Дан многочлен $P(x_1, \dots, x_n)$ от нескольких переменных с целыми коэффициентами. Спрашивается, есть у него целочисленное решение, т.е. такой набор целых чисел $a_1, \dots, a_n$, что $P(a_1, \dots, a_n) = 0$?