Модели вычислений

В.А. Захаров, Р.И. Подловченко

Лекция 6.

1. Проблема соответствий Поста
2. Многоголовочные конечные автоматы
3. Ассоциативные исчисления. Системы Туэ
4. Другие алгоритмически неразрешимые математические проблемы

АЛГОРИТМИЧЕСКИ НЕРАЗРЕШИМЫЕ МАТЕМАТИЧЕСКИЕ ЗАДАЧИ

Мы рассмотрим несколько известных математических задач из разных областей — комбинаторики, алгебры, теории автоматов, — и докажем их алгоритмическую неразрешимость.

Для каждой из этих задач ее алгоритмическая неразрешимость доказывается одним и тем же способом: сведением к этой задаче проблемы останова машины Тьюринга на пустой ленте. После этого становится понятно, какое большое значение для развития математики имеет теорема о неразрешимости проблемы останова, установленная Аланом Тьюрингом.

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Пусть имеется некоторый конечный алфавит Σ и конечное множество пар слов в этом алфавите

$$\mathcal{P} = \{(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)\}.$$

Проблема соответствий Поста (ПСП) состоит в том, чтобы выяснить, существует ли такая конечная последовательность индексов $i_1, i_2, \ldots, i_n$, для которой выполняется равенство

$$u_{i_1} u_{i_2} \ldots u_{i_n} = v_{i_1} v_{i_2} \ldots v_{i_n}$$

Указанная последовательность индексов $i_1, i_2, \ldots, i_n$ называется решением проблемы соответствий Поста для системы пар $\mathcal{P}$.

Проблема соответствий Поста

Пример 1. Проблема соответствий Поста для системы пар $\mathcal{P} = \{(b, babbb), (ab, abba), (abb, \varepsilon)\}$ имеет решение $3, 2, 3, 1$, поскольку

$$abb\ ab\ abb\ b = \varepsilon\ abba\ \varepsilon babbb$$

Пример 2. Проблема соответствий Поста для системы пар $\mathcal{P}_2 = \{(b, bab), (ab, aba), (abb, bba)\}$ не имеет решений. (Почему?)

Нас будет интересовать вопрос о существовании алгоритма, который для произвольной заданной системы пар $\mathcal{P}$ устанавливает, имеет ПСП решение или нет.

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Содержательный смысл ПСП таков. Предположим, что есть два алфавитных кодирования одного и того же алфавита сообщений

$$\begin{array}{llll} \varphi: & 1 \to u_1, & \dots & k \to u_k; \\ \psi: & 1 \to v_1, & \dots & k \to v_k. \end{array}$$

Вопрос: существует ли такое сообщение $i_1, i_2, \dots, i_n$, которое кодируется одинаково системами кодирования φ и ψ, т.е.

$$\varphi(i_1 i_2 \dots i_n) = u_{i_1} u_{i_2} \dots u_{i_n} = v_{i_1} v_{i_2} \dots v_{i_n} = \psi(i_1 i_2 \dots i_n)?$$

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Проблема соответствий Поста называется инициальной (ИПСП) в том случае, если дополнительно требуется, чтобы ее решение начиналось индексом 1 .

Утверждение 6.1. ИПСП алгоритмически сводима к ПСП.

Доказательство. Покажем, что для любой системы пар $\mathcal{P}$ можно построить такую систему пар $\widehat{\mathcal{P}}$, что ИПСП для $\mathcal{P}$ имеет решение тогда и только тогда, когда ПСП для $\widehat{\mathcal{P}}$ имеет решение.

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Пусть $\mathcal{P} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$. Рассмотрим две новые буквы $\#, \$$, и для каждой пары слов $u_i = x_1 x_2 \ldots x_p$, $v_i = y_1 y_2 \ldots y_q$,
определим новую пару слов
$\widehat{u}_i = x_1 \# x_2 \# \ldots \# x_p \#$, $\widehat{v}_i = \# y_1 \# y_2 \# \ldots \# y_q$.
Кроме того, построим еще две пары слов
$\widehat{u}_0 = \# \widehat{u}_1$, $\widehat{v}_0 = \widehat{v}_1$,
$\widehat{u}_{k+1} = \$$, $\widehat{v}_{k+1} = \#\$$,
и положим
$\widehat{\mathcal{P}} = \{(\widehat{u}_0, \widehat{v}_0), (\widehat{u}_1, \widehat{v}_1), \ldots, (\widehat{u}_k, \widehat{v}_k), (\widehat{u}_{k+1}, \widehat{v}_{k+1})\}$.

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Если последовательность индексов
$1, i_2, i_3, \ldots, i_n$
является решением ИПСП для $\mathcal{P}$, т.е.

$$u_1 u_{i_1} u_{i_2} \ldots u_{i_n} = v_1 v_{i_1} v_{i_2} \ldots v_{i_n},$$

то последовательность индексов
$0, i_2, i_3, \ldots, i_n, k+1$
является решением ПСП для $\widehat{\mathcal{P}}$, т.е.

$$\# \widehat{u}_1 \widehat{u}_{i_1} \widehat{u}_{i_2} \ldots \widehat{u}_{i_n} \$ = \widehat{v}_1 \widehat{v}_{i_1} \widehat{v}_{i_2} \ldots \widehat{v}_{i_n} \# \$.$$

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Пусть последовательность индексов $j_1, j_2, \ldots, j_m$ является <u>кратчайшим</u> решением ПСП для $\widehat{\mathcal{P}}$, т.е.

$$\widehat{u}_{j_1} \widehat{u}_{j_2} \ldots \widehat{u}_{j_m} = \widehat{v}_{j_1} \widehat{v}_{j_2} \ldots \widehat{v}_{j_m}.$$

Тогда $j_1 = 0$, поскольку только пара слов $\widehat{u}_0, \widehat{v}_0$ начинается одной и той же буквой $\#$,
$j_m = k+1$, поскольку только пара слов $\widehat{u}_{k+1}, \widehat{v}_{k+1}$ оканчивается одной и той же буквой $\$$,
индекс $k+1$ не встречается среди $j_2, \ldots, j_{m-1}$, поскольку рассматриваемое решение кратчайшее,
индекс 0 не встречается среди $j_2, \ldots, j_{m-1}$, поскольку в слове $\widehat{v}_{j_1} \widehat{v}_{j_2} \ldots \widehat{v}_{j_m}$ нет идущих подряд букв $\#$.

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Таким образом из равенства

$$\widehat{u}_{j_1}\widehat{u}_{j_2}\dots\widehat{u}_{j_m} = \widehat{v}_{j_1}\widehat{v}_{j_2}\dots\widehat{v}_{j_m}$$

следует, что

$$\#\widehat{u}_1\widehat{u}_{j_2}\dots\widehat{u}_{j_{m-1}}\$ = \widehat{v}_{j_1}\widehat{v}_{j_2}\dots\widehat{v}_{j_{m-1}}\#\$$$

а отсюда следует равенство

$$u_1 u_{j_2}\dots u_{j_{m-1}} = v_1 v_{j_2}\dots v_{j_{m-1}},$$

т.е. ИПСП для $\mathcal{P}$ имеет решение $j_2, \dots, j_{m-1}$.

QED

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Утверждение 6.2. Проблема останова машин Тьюринга алгоритмически сводима к ИПСП.

Доказательство. Покажем, что для любой МТ $\mathcal{M}$ можно построить такую систему пар $\mathcal{P}_{\mathcal{M}}$, что ИПСП для $\mathcal{P}_{\mathcal{M}}$ имеет решение тогда и только тогда, когда $\mathcal{M}$ останавливается на пустой ленте.

Идея сведения такова .

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Предположим, что МТ $\mathcal{M}$ имеет конечное вычисление на пустой ленте

$$\alpha_0 = q_1 0 \longrightarrow \alpha_1 \longrightarrow \cdots \longrightarrow \alpha_N = w' q_1 w''$$

Построим ИПСП $\mathcal{P}_{\mathcal{M}} = \mathcal{P}_1 \cup \mathcal{P}_2$, которая имеет решение, порождающее пару одинаковых слов вида

$$\underbrace{\$\alpha_0\$\alpha_1\$\ldots\$\alpha_N\$}_{\text{симуляция вычисления}} \quad \overbrace{\ldots\$q_0 w''\$\ldots\$q_0\$\$}^{\text{стирание результата}}$$

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Множество $\mathcal{P}_1$ образуют следующие пары слов, позволяющие моделировать вычисление МТ $\mathcal{M}$:

1. $u_1 = \$$, $v_1 = \$q_0 0\$$. Эта пара всегда будет первой в любом решении;
2. $u_i = c$, $v_i = c$ для всех букв ленточного алфавита;
3. $u_i = \$$, $v_i = \$$;
4. $u_i = cq'a$, $v_i = q''cb$ для каждой команды вида $\langle q'a : bq''L\rangle$ МТ $\mathcal{M}$ и ленточной буквы c;
5. $u_i = \$q'a$, $v_i = \$q''0b$ для каждой команды вида $\langle q'a : bq''L\rangle$ МТ $\mathcal{M}$;
6. $u_i = q'ac$, $v_i = bq''c$ для каждой команды вида $\langle q'a : bq''R\rangle$ МТ $\mathcal{M}$ и ленточной буквы c;
7. $u_i = q'a\$$, $v_i = bq''0\$$ для каждой команды вида $\langle q'a : bq''R\rangle$ МТ $\mathcal{M}$.

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

$\$q_1 0\$\alpha_2\$\alpha_3\$\ldots\ldots\$\alpha_{N-1}\$$

$\$q_1 0\$\alpha_2\$\alpha_3\$\alpha_4\$\ldots\$\alpha_{N-1}\$w'q_0w''\$$

Построение решения ИПСП начинается с пары (u_1, v_1).
А дальше в стремлении сделать первую строку равной второй строке мы вынуждены будем выбирать пары из множества $\mathcal{P}_1$ однозначно, моделируя тем самым вычисление МТ $\mathcal{M}$.
И процесс этот может остановиться тогда и только тогда, когда во второй строке будет достигнута заключительная конфигурация $\alpha_N = w'q_0w''$.

ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

$$\$q_1 0\$\alpha_2\$\alpha_3\$\dots\dots\$\alpha_{N-1}\$w'q_0w''\$\dots\$q_0\$\$$$

$$\$q_1 0\$\alpha_2\$\alpha_3\$\alpha_4\$\dots\$\alpha_{N-1}\$w'q_0w''\$\dots\$q_0\$\$$$

После того, как во второй строке была достигнута заключительная конфигурация, нужно суметь выровнять обе строки.
И для этого достаточно «стирать» поочередно все ленточные буквы при помощи следующих пар слов из множества $\mathcal{P}_2$

8 $u_i = aq_0$, $v_i = q_0$ для каждой ленточной буквы a ;

9 $u_i = \$q_0a$, $v_i = \$q_0$ для каждой ленточной буквы a ;

10 $u_i = q_0\$\$$, $v_i = \$$.

В результате обе строки становятся одинаковыми.

QED

Проблема соответствий Поста

Теорема 6.3. Проблема соответствий Поста алгоритмически неразрешима.

Доказательство. Из Утверждений 6.1, 6.2. можно сделать вывод о том, что существует алгоритм, который для любой МТ $\mathcal{M}$ строит такую систему пар $\widehat{\mathcal{P}}_{\mathcal{M}}$, что

$\mathcal{M}$ останавливается на пустой конфигурации $\Leftrightarrow$ ПСП для $\widehat{\mathcal{P}}_{\mathcal{M}}$ имеет решение.

Таким образом, из алгоритмической неразрешимости проблемы останова машин Тьюринга следует неразрешимость ПСП. QED

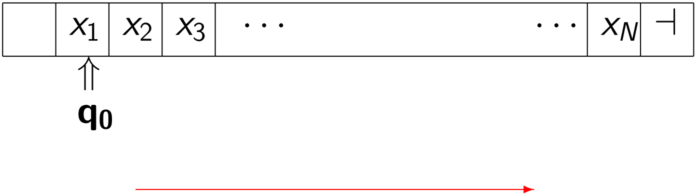
ПРОБЛЕМА СООТВЕТСТВИЙ ПОСТА

Задача 1. Будет проблема соответствий Поста оставаться алгоритмически неразрешимой, если алфавит Σ, в котором задаются слова всех пар системы $\mathcal{P}$, состоит из одной буквы?

А что будет, если этот алфавит состоит из двух букв?

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

Конечный автомат представляет собой устройство, прочитывающее слово, записанное на ленте.
Обычный конечный автомат имеет только одну считывающую головку, которая движется по ленте в одну сторону.



Многоголовочные конечные автоматы

Если автомату разрешить сдвигать считывающую головку в обе стороны...

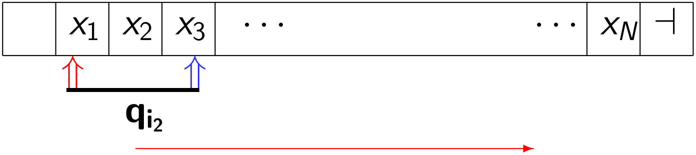
	x_1	x_2	x_3	$\cdots \qquad \cdots$	x_N	$\dashv$

$$\Uparrow$$
$$\mathbf{q_0}$$

то, как это ни удивительно, его вычислительные возможности не возрастут.

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

А что произойдет, если автомат будет иметь **ДВЕ** считывающие головки, которые могут перемещаться только в одну сторону, но при этом независимо друг от друга?



Оказывается, что вычислительные возможности такого автомата усиливаются настолько, что он способен моделировать машины Тьюринга!

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

Покажите самостоятельно, что двухголовочный автомат способен распознавать следующие языки, не являющиеся регулярными:

1. $L_1 = \{a^n b^n \ : \ n \geq 0\}$;
2. $L_2 = \{a^n b^n c^n \ : \ n \geq 0\}$;
3. $L_3 = \{ww \ : \ w \in \{a, b\}^*\}$;

А способен ли двухголовочный автомат распознавать язык $L_4 = \{ww^- \ : \ w \in \{a, b\}^*\}$?

Оказывается, что нет!

Задача 2. Доказать, что язык L_4 не распознается двухголовочными автоматами.

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

Почему же двухголовочный автомат способен моделировать машины Тьюринга?

Потому что верна

Теорема 6.4. Проблема пустоты языков, распознаваемых двухголовочными автоматами, алгоритмически неразрешима.

Но прежде чем ее доказывать, дадим формальные определения двухголовочных автоматов и их вычислений.

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

Двухголовочный детерминированный конечный автомат — это система $\mathcal{D} = (\Sigma, S, s_0, F, T)$, где

- Σ — ленточный алфавит,
- S — конечное множество состояний,
- s_0 — начальное состояние,
- F — подмножество финальных состояний,
- $T : S \times (\Sigma \cup \{\dashv\}) \times (\Sigma \cup \{\dashv\}) \to S \times \{0, 1\} \times \{0, 1\}$ — функция переходов.

Здесь $T(s, x_1, x_2) = (s', \delta_1, \delta_2)$ означает, что если автомат пребывает в состоянии s, и его считывающие головки обозревают на ленте буквы или граничный маркер x_1 и x_2, то на следующем такте вычисления он переходит в состояние s, и его считывающие головки сдвигаются вправо на δ_1 и δ_2 ячеек.

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

w -конфигурацией двухголовочного автомата называется набор вида $(w \dashv, s, i_1, i_2)$, где

- $w = x_1 x_2 \ldots x_n \in \Sigma^*$ — слово, записанное на ленте;
- $s \in S$ — текущее состояние автомата,
- i_1, i_2 — номера букв (ячеек ленты), обозреваемых считывающими головками автомата, $1 \leq i_1, i_2 \leq n + 1$.

Конфигурация $(w \dashv, s_0, 1, 1)$ считается начальной

.

Всякая конфигурация вида $(w \dashv, s, n + 1, n + 1)$, где $s \in F$, считается финальной

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

Преобразование конфигураций $(w \dashv, s, i_1, i_2) \longrightarrow (w \dashv, s', j_1, j_2)$ автоматом $\mathcal{D}$ выполняется в том и только том случае, если

- $w = x_1 x_2 \dots x_n$,
- $T(s, x_{i_1}, x_{i_2}) = (s', \delta_1, \delta_2)$,
- $j_1 = i_1 + \delta_1, \ j_2 = i_2 + \delta_2$.

Вычислением $conf_1 \longrightarrow_* conf_N$ двухголовочного конечного автомата $\mathcal{D}$ на слове w называется всякая последовательность преобразований w-конфигураций

$$conf_1 \longrightarrow conf_2 \longrightarrow \cdots \longrightarrow conf_N .$$

Это вычисление считается успешным, если $conf_0$ — начальная, а $conf_N$ — финальная конфигурации.
Язык двухголовочного автомата $\mathcal{D}$ — это множество слов $L(\mathcal{D}) = \{w :$ существует успешное w – вычисление автомата $\mathcal{D}\}$.

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

Утверждение 6.5. Проблема останова машин Тьюринга алгоритмически сводима к проблеме пустоты двухголовочных детерминированных конечных автоматов.

Доказательство. Покажем, что для любой МТ $\mathcal{M}$ можно построить такой двухголовочный автомат $\mathcal{D}_{\mathcal{M}}$, что
$L(\mathcal{P}_{\mathcal{M}}) \neq \varnothing \Leftrightarrow \mathcal{M}$ останавливается на пустой ленте.

Идея сведения такова .

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

Предположим, что МТ $\mathcal{M}$ имеет вычисление на пустой ленте

$$\alpha_0 = q_1 0 \longrightarrow \alpha_1 \longrightarrow \alpha_2 \longrightarrow \cdots\cdots$$

Построим двухголовочный автомат $\mathcal{D}_{\mathcal{M}}$, язык $L(\mathcal{P}_{\mathcal{M}})$ которого либо содержит только одно слово

$$w = \$\alpha_0\$\alpha_1\$\ldots\$\alpha_N,$$

если указанное вычисление МТ $\mathcal{M}$ завершается спустя N тактов, либо является пустым, если указанное вычисление МТ $\mathcal{M}$ бесконечно.

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

Сценарий работы автомата $\mathcal{D}_{\mathcal{M}}$ таков.

1. Начав работать с w -конфигурации

$$\underset{\Uparrow_1 \Uparrow_2}{\$}\ \alpha_0\ \$\ \alpha_1\ \$\ \dots\ \$\ \alpha_{N-1}\ \$\ \alpha_N\ \dashv,$$

автомата $\mathcal{D}_{\mathcal{M}}$ продвигает считывающую головку $\Uparrow_1$ до следующего разделителя $\$$ и при этом проверяет, что слово α_0, расположенное между этими разделителями — это начальная конфигурация $\alpha_0 = q_1 0$ МТ:

$$\underset{\Uparrow_2}{\$}\ \alpha_0\ \underset{\Uparrow_1}{\$}\ \alpha_1\ \$\ \dots\ \$\ \alpha_{i-1}\ \$\ \alpha_i\ \$\ \dots\ \$\ \alpha_{N-1}\ \$\ \alpha_N\ \dashv,$$

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

2. Затем автомат $\mathcal{D}_{\mathcal{M}}$, синхронно перемещая обе считывающие головки, сравнивает побуквенно слова α_0 и α_1 и проверяет, что α_1 — это конфигурация МТ $\mathcal{M}$, образующаяся за один такт ее работы из конфигурации α_0:

$$\underset{\Uparrow_2}{\$}\ \alpha_0\ \underset{\Uparrow_1}{\$}\ \alpha_1\ \$\ \ldots\ \$\ \alpha_{i-1}\ \$\ \alpha_i\ \$\ \ldots\ \$\ \alpha_{N-1}\ \$\ \alpha_N\ \dashv,$$

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

3. Это синхронное перемещение считывающих головок и побуквенное сравнение последовательно идущих фрагментов α_{i-1} и α_i для подтверждения того, что α_i получена из α_{i-1} за один такт работы МТ $\mathcal{M}$, проводится для всех i, $i \geq 1$.

$$\$\ \alpha_0\ \$\ \alpha_1\ \$\ \ldots\ \underset{\Uparrow 2}{\$}\ \alpha_{i-1}\ \underset{\Uparrow 1}{\$}\ \alpha_i\ \$\ \ldots\ \$\ \alpha_{N-1}\ \$\ \alpha_N\ \dashv,$$

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

4. Но как только считывающая головка $\Uparrow_1$ при проверке очередной пары фрагментов α_{N-1} и α_N обнаруживает, что фрагмент $\alpha_N = uq_0v$ — это заключительная конфигурация МТ $\mathcal{M}$

$$\$\ \alpha_0\ \$\ \alpha_1\ \$\ \ldots\ \$\ \alpha_{i-1}\ \$\ \alpha_i\ \$\ \ldots\ \$\ \underset{\Uparrow_2}{\alpha_{N-1}}\ \$\ \underset{\substack{\Uparrow_1 \\ \text{«вижу } q_0\text{»}}}{\alpha_N}\ \dashv,$$

двухголовочный автомат $\mathcal{D}_{\mathcal{M}}$ переходит в режим успешного завершения вычисления.

МНОГОГОЛОВОЧНЫЕ КОНЕЧНЫЕ АВТОМАТЫ

5. Для успешного завершения вычисления автомат $\mathcal{D}_{\mathcal{M}}$ проверяет, что считывающая головка $\Uparrow_1$ достигает по прочтении фрагмента α_N граничного маркера $\dashv$

$$\$\ \alpha_0\ \$\ \alpha_1\ \$\ \ldots\ \$\ \alpha_{i-1}\ \$\ \alpha_i\ \$\ \ldots\ \$\ \alpha_{N-1}\ \$\ \alpha_N\ \underset{\Uparrow_2\Uparrow_1}{\dashv},$$

а затем перемещает к этому маркеру считывающую головку $\Uparrow_2$ и переходит в финальное состояние.

QED

Многоголовочные конечные автоматы

Из доказанного Утверждения 6.5 и теоремы об алгоритмической неразрешимости проблемы останова для машин Тьюринга следует Теорема 6.4 об алгоритмической неразрешимости проблемы пустоты для двухголовочных детерминированных конечных автоматов.

Ассоциативные исчисления

В 1912 г. норвежский математик Аксель Туэ предложил одну из самых простых моделей вычислений. Впоследствие А.А. Марков предложил называть системы вычислений такого вида **ассоциативными исчислениями**, поскольку они относятся к алгебраическим системам с одной операцией, подчиняющейся единственному закону — закону ассоциативности.

Фактически, ассоциативное исчисление — это набор правил переписывания слов путем замены одних подслов другими. Но даже для такой простой модели вычислений некоторые задачи оказываются алгоритмически неразрешимыми.

Ассоциативные исчисления

Пусть задан конечный алфавит $\Sigma = \{a_1, a_2, \ldots, a_n\}$ и конечное множество упорядоченных пар слов в этом алфавите $\mathcal{K} = \{(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)\}$.

Определим правило вывода слов следующим образом: слово $\widehat{w}$ непосредственно выводимо из слова w в системе $\mathcal{K}$ (обозначается $w \xrightarrow{\mathcal{K}} \widehat{w}$), если существует такое i, $1 \leq i \leq k$, , для которого $w = w' u_i w''$ и $\widehat{w} = w' v_i w''$.

АССОЦИАТИВНЫЕ ИСЧИСЛЕНИЯ

Слово w_m считается выводимым из слова w_0 в системе $\mathcal{K}$, если существует конечная последовательность непосредственно выводимых друг из друга слов

$$w_0 \xrightarrow{\mathcal{K}} w_1 \xrightarrow{\mathcal{K}} w_2 \xrightarrow{\mathcal{K}} \cdots \xrightarrow{\mathcal{K}} w_{m-1} \xrightarrow{\mathcal{K}} w_m.$$

Отношение выводимости в системе $\mathcal{K}$ будем обозначать $w_0 \xrightarrow{\mathcal{K}}_* w_m$.

Система упорядоченных пар слов $\mathcal{K}$ с введенным отношением непосредственной выводимост$\xrightarrow{\mathcal{K}}$ называется полусистемой Туэ .

Ассоциативные исчисления

Пример. Пусть $\mathcal{K} = \{(abb, bab), (aba, b)\}$. Тогда

$$aabbaa \xrightarrow{\mathcal{K}} ababaa \xrightarrow{\mathcal{K}} abba \xrightarrow{\mathcal{K}} baba \xrightarrow{\mathcal{K}} bb$$

АССОЦИАТИВНЫЕ ИСЧИСЛЕНИЯ

Полусистемы Туэ послужили первоосновой для многих других математических моделей — машин Тьюринга, нормальных алгоритфмов Маркова, формальных грамматик и пр.

Для полусистем Туэ интерес представляет следующая задача — проблема выводимости :

для заданной полусистемы Туэ $\mathcal{K}$ и пары слов w' и w'' выяснить, выводимо ли слово w'' из w' слова в $\mathcal{K}$.

Ассоциативные исчисления

Теорема 6.6. Проблема выводимости для полусистем Туэ алгоритмически неразрешима.

Доказательство этой теоремы основывается на следующем утверждении.

Утверждение 6.7. Для любой машины Тьюринга $\mathcal{M}$ можно эффективно построить такую полусистему Туэ $\mathcal{K}_{\mathcal{M}}$ и такую пару слов w', w'', что слово w'' выводимо из слова w' в $\mathcal{K}$ в том и только том случае, когда МТ $\mathcal{M}$ останавливается на пустой ленте.

АССОЦИАТИВНЫЕ ИСЧИСЛЕНИЯ

Доказательство. Пусть МТ $\mathcal{M}$ имеет множество состояний $Q = \{q_0, q_1, \ldots, q_m\}$ и ленточный алфавит $\{0, 1\}$. Полусистема Туэ $\mathcal{K}_{\mathcal{M}}$ включает следующие пары слов в алфавите $\{0, 1, \#, q_0, q_1, \ldots, q_m\}$:

- для каждой команды $\langle q, a : q', b, L\rangle$ три пары $(0qa, q'0b)$, $(1qa, q'1b)$ и $(\#qa, \#q'0b)$;
- для каждой команды $\langle q, a : q', b, R\rangle$ три пары $(qa0, bq'0)$, $(qa1, bq'1)$ и $(qa\#, bq'0\#)$;
- четыре пары $(q_0 0, q_0)$, $(q_0 1, q_0)$, $(0q_0, q_0)$ и $(1q_0, q_0)$.

В качестве w', w'' возьмем слова $\#q_1 0\#$ и $\#q_0\#$.

АССОЦИАТИВНЫЕ ИСЧИСЛЕНИЯ

Как видно из определения полусистемы $\mathcal{K}_{\mathcal{M}}$, для каждой незаключительной конфигурации α МТ $\mathcal{M}$ из слова $\#\alpha\#$ непосредственно выводимо слово w тогда и только тогда, когда МТ $\mathcal{M}$ преобразует за один такт работы конфигурацию α в конфигурацию β и при этом $w = \#\beta\#$.

Поэтому в полусистеме Туэ $\mathcal{K}_{\mathcal{M}}$ из слова $w' = \#q_1 0\#$ выводимо слово, содержащее символ q_0 тогда и только тогда, когда МТ $\mathcal{M}$ имеет успешное вычисление из начальной конфигурации $\alpha_0 = q_1 0$.

АССОЦИАТИВНЫЕ ИСЧИСЛЕНИЯ

Нетрудно видеть также, что для каждой заключительной конфигурации $\beta = uq_0v$ из слова $\#uq_0v\#$ в полусистеме Туэ $\mathcal{K}_{\mathcal{M}}$ выводимо $w'' = \#q_0\#$.

Таким образом, полусистеме Туэ $\mathcal{K}_{\mathcal{M}}$ из слова $w' = \#q_1 0\#$ выводимо слово $w'' = \#q_0\#$ тогда и только тогда, когда МТ МТ $\mathcal{M}$ останавливается на пустой ленте.

QED

АССОЦИАТИВНЫЕ ИСЧИСЛЕНИЯ

Определение непосредственной выводимости слов можно усилить: слово $\widehat{w}$ непосредственно выводимо из слова w в системе $\mathcal{K}$ (обозначается $w \overset{\mathcal{K}}{\Longrightarrow} \widehat{w}$), если существует такое i, $1 \leq i \leq k$, , для которого либо $w = w' u_i w''$ и $\widehat{w} = w' v_i w''$, либо $w = w' v_i w''$ и $\widehat{w} = w' u_i w''$.

Таким образом, пары (u_i, v_i) — это тождества $u_i = v_i$, а вывод $w \overset{\mathcal{K}}{\Longrightarrow}_* \widehat{w}$ — это цепочка тождественных преобразований слов.

Система упорядоченных пар слов $\mathcal{K}$ с введенным отношением непосредственной выводимости $\overset{\mathcal{K}}{\Longrightarrow}$ называется системой Туэ.

АССОЦИАТИВНЫЕ ИСЧИСЛЕНИЯ

Задача 3. Доказать, что существует такая полусистема Туэ $\mathcal{K}$, для которой алгоритмически неразрешима следующая задача: для заданной пары слов w' и w'' выяснить, выводимо ли слово w'' из w' слова в $\mathcal{K}$.

В 1967 г. Ю.В. Матиясевич построил такую систему, содержащую всего лишь три пары слов.

Задача 4. [Трудная] Доказать, что алгоритмически неразрешима следующая задача: для заданной системы Туэ $\mathcal{K}$ и пары слов w' и w'' выяснить, выводимо ли слово w'' из w' слова в $\mathcal{K}$.

АЛГОРИТМИЧЕСКИ НЕРАЗРЕШИМЫЕ ЗАДАЧИ

Существуют и другие алгоритмически неразрешимые математические задачи.

Рассмотрим наиболее известные из них.

Диофантовы уравнения

Уравнение $P(x_1, x_2, \ldots, x_n) = 0$, где $P(x_1, x_2, \ldots, x_n)$ — многочлен с целочисленными коэффициентами, называют диофантовым уравнением. Названы в честь древнегреческого математика Диофанта («отец алгебры»).

Примером диофантова уравнения является любое из уравнений вида $x^n + y^n = z^n$.

ДИОФАНТОВЫ УРАВНЕНИЯ

Многие столетия математики искали способ получения целочисленных решений диофантовых уравнений. Например, для диофантовых уравнений с одной переменной такой способ предоставляет теорема Безу.

В 1900 г. Давид Гильберт сформулировал задачу (10-я проблема Гильберта)

Пусть задано диофантово уравнение с произвольными неизвестными и целыми рациональными числовыми коэффициентами. Указать способ, при помощи которого возможно после конечного числа операций установить, разрешимо ли это уравнение в целых рациональных числах.

ДИОФАНТОВЫ УРАВНЕНИЯ

В 1953-1970 гг. М. Девис, Ю. Робинсон, Х. Патнем и Ю.В. Матиясевич доказали алгоритмическую неразрешимость этой проблемы. Окончательная формулировка полученного результата такова.

Теорема. Для любого рекурсивно перечислимого множества целых чисел R существует такой многочлен $P(x_1, \ldots, x_n, z)$, что для любого целого числа k справедливо следующее соотношение

$k \in R$

$\Leftrightarrow$

диофантово уравнение $P(x_1, \ldots, x_n, k) = 0$ имеет целочисленное решение.

МАТРИЧНАЯ АЛГЕБРА

Предположим, что имеется некоторое конечное множество целочисленных матриц $M_1, M_2, \ldots, M_k$ размера $n \times n$.

Зададимся вопросом (проблема единичной матрицы): можно ли, перемножая матрицы этого множества, получить единичную матрицу I, т.е. сформировать такую конечную последовательность индексов $i_1, i_2, \ldots, i_N$, чтобы выполнялось равенство

$$M_{i_1} \times M_{i_2} \times \cdots \times M_{i_N} = I?$$

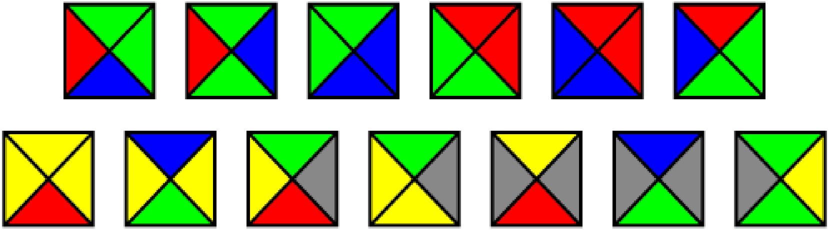
МАТРИЧНАЯ АЛГЕБРА

Теорема. Проблема единичной матрицы алгоритмически неразрешима для целочисленных матриц размера $n \times n$ при любых $n, n \geq 4$, и разрешима для $n = 2$.

Открытая проблема. Существует ли алгоритм решения проблема единичной матрицы для матриц размера 3×3?

ПРОБЛЕМА МОЗАИКИ

Мозаика (домино): — это конечное множество (образцов) квадратных плиток, края которых могут иметь разнообразную раскраску.

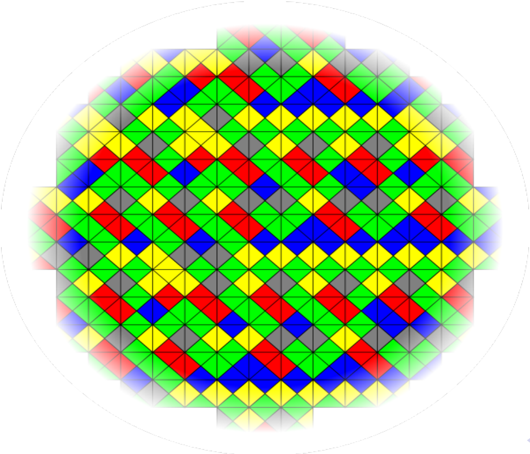


Две плитки, прилегающие друг к другу краями, считаются совместными, если их смежные края одинаково окрашены.

При этом вращать плитки запрещается.

Проблема мозаики

Зададимся вопросом (проблема мозаики): можно ли покрыть плитками заданной мозаики всю бесконечную плоскость так, чтобы любая пара соседних плиток была совместной?



Проблема мозаики

Теорема. Проблема мозаики алгоритмически неразрешима.

Открытая проблема. Существует ли алгоритм решения проблема единичной матрицы для матриц размера 3×3?

Машины Минского

Насколько простой может быть модель вычислений, дающая возможность реализовать любой алгоритм (вычислить любую рекурсивную функцию)?

Кажется, что трудно представить себе универсальное вычислительное устройство, более простое, чем машины Тьюринга.

Однако есть и более простые универсальные модели вычислений!

Машины Минского

Машина Минского — это программа, состоящая из помеченных операторов только двух видов:

- L': X++; **goto** L'' ;
- L': **if** X>0 **then** X−− ; **goto** L'' **else goto** L''' .

Переменные принимают только целые неотрицательные значения.

Машину Минского можно представить в виде автомата, снабженного набором счетчиков. На каждом такте работы показания какого-нибудь счетчика либо увеличиваются, либо (после проверки его непустоты) уменьшаются.

Теорема. Для любой частично-рекурсивной функции существует машина Минского с двумя счетчиками, вычисляющая эту функцию.

Следствие. Проблема останова для машины Минского с двумя счетчиками алгоритмически неразрешима.

Задача 5. Докажите вычислительную универсальность машин Минского с тремя счетчиками.

Задача 6. Постройте алгоритм решения проблемы останова для машин Минского с одним счетчиком.

КОНЕЦ ЛЕКЦИИ 6