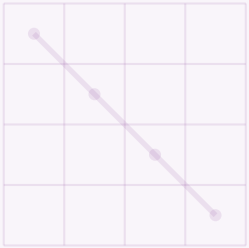
11

Лекция 20–21 апреля

Проблема остановки

Константин Чухарев

Дискретная математика — Весна 2027



Диагональный аргумент

> *Сущность математики лежит именно в её свободе.*
>
> ***— Георг Кантор***

Диагональ Кантора

В первом семестре диагональный метод доказал несчётность действительных чисел.

Дан список чисел $r_1, r_2, r_3, \ldots$ — строим новое число, меняя n-ю цифру на отличную от n-й цифры r_n.

Новое число отличается от каждого элемента списка в своей позиции — списка всех действительных чисел не существует.

> Тот же приём убьёт мечту об универсальном анализаторе программ.
>
> Диагонализация везде устроена одинаково: объект, отличающийся от любого элемента перечисления, обязан в нём состоять.

Инверсия диагонали

Предположим, каждая машина останавливается на каждой кодировке.

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$
M_1	ост.	ост.	цикл	ост.
M_2	цикл	цикл	ост.	ост.
M_3	ост.	цикл	цикл	цикл
M_4	ост.	ост.	ост.	цикл
D — инверсия	цикл	ост.	ост.	ост.

Строка D отличается от каждой строки списка — но D сама машина и обязана быть в списке.

Вопрос $H(\langle M \rangle, \langle M \rangle)$ — «остановится ли M на своём коде»: диагональ выбирает позицию, где ответ обязан противоречить самому себе.

Самоприменимость законна

Кодировка $\langle M \rangle$ — такая же строка, как любая другая, а машина умеет читать строки.

Подать машине её собственное описание — не страннее, чем передать функции её номер.

После универсальной машины это уже не сюрприз: программа читает программу штатно.

Диагонализация требует лишь одного — выразительности, достаточной для разговора о себе.

Проблема остановки

> *Математику можно определить как предмет, в котором мы никогда не знаем ни того, о чём говорим, ни того, истинно ли то, что мы говорим.*
>
> **— Бертран Рассел**

Проблема остановки

Определение 1 («HALT»)

По паре $(\langle M \rangle, w)$ узнать, останавливается ли M на входе w:

$$\mathrm{HALT} = \{\langle M \rangle w \mid M \text{ останавливается на } w\}.$$

Пример (Конкретные запросы)

- M_{loop} — на любом входе сразу зацикливается: $\langle M_{\mathrm{loop}} \rangle 0 \notin \mathrm{HALT}$.
- M_{acc} — на любом входе сразу принимает: $\langle M_{\mathrm{acc}} \rangle \varepsilon \in \mathrm{HALT}$.

Тривиальные случаи решаются локально — вопрос в одном универсальном алгоритме для всех пар.

Почему симуляция не решает?

Наивный план: запустить U на $\langle M \rangle w$ и подождать ответа.

Если M останавливается — симуляция увидит это.

Если M зациклилась, ждать можно вечно.

Любой конечный таймаут произволен: зациклившаяся машина может остановиться на секунду позже.

Симулятор лишь воспроизводит проблему — нужен другой решатель, и его не существует.

Неразрешимость остановки

Теорема 1 (Неразрешимость HALT)

Не существует МТ, которая по паре $(\langle M \rangle, w)$ всегда останавливается и решает, остановится ли M на w.

Доказательство

Докажем от противного диагонализацией.

Пусть H — решатель: всегда останавливается и принимает ровно на останавливающихся парах $\langle M \rangle w$.

Построим D: на $\langle M \rangle$ она спрашивает H про пару $(\langle M \rangle, \langle M \rangle)$ и делает наоборот — при приёме зацикливается, при отвержении останавливается.

Случай 1: D останавливается на $\langle D \rangle$. Тогда H отверг — и D обязана зациклиться.
Случай 2: D зацикливается на $\langle D \rangle$. Тогда H принял — и D обязана остановиться.

Распознаваема, но не разрешима

HALT распознаваема: универсальная машина симулирует M на w и принимает при остановке.

Но не разрешима — доказано.

Дополнение $\overline{\text{HALT}}$ — пары с зацикливанием — даже не распознаваемо.

Иначе, запустив обе машины параллельно и чередуя шаги, получали бы ответ на любой паре — а это запрещено теоремой.

Полная характеристика разрешимости через две распознаваемости ждёт на следующей лекции.

Одно семейство доказательств

Объект	Самореференция	Полученное «нет»
Лжец	$S \leftrightarrow \neg S$	нет истинностного значения
Рассел	$R = \{x \mid x \notin x\}$	нет множества
Кантор	диагональ вне списка	нет перечисления
Гёдель	«недоказуемо»	истина вне вывода
Тьюринг	D на своём коде	нет алгоритма

Система, достаточно выразительная, чтобы говорить о себе, порождает неразрешимость.

Неразрешимость — не наша неспособность придумать алгоритм, а свойство выразительности.

Сведения

> *Если не удаётся решить предложенную задачу,*
> *попробуйте сначала решить какую-нибудь родственную.*
>
> ***— Дьёрдь Пойа***

Сведение

Определение 2 (Сведение (*reduction*))

$A \underset{m}{\leq} B$, если существует вычислимая тотальная функция f, для которой

$$w \in A \leftrightarrow f(w) \in B.$$

Решив B, решаем и A: считаем $f(w)$ и спрашиваем решатель B.

Сведение переносит алгоритмы — и неразрешимости.

Как пользоваться сведением?

Теорема 2 (Логика переноса)

- Если $A \leq_m B$ и B разрешима, то разрешима и A.
- Контрапозитивно: если $A \leq_m B$ и A неразрешима, то неразрешима и B.

Сводить A к разрешимой B бессмысленно — из этого следует лишь разрешимость A.

Чтобы доказать неразрешимость задачи X, сводят к ней известную неразрешимую: $\mathrm{HALT} \leq_m X$.

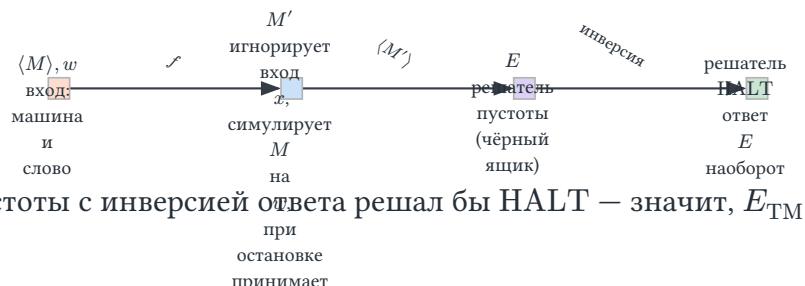
Пример: пустота языка

Пример (От HALT к E_{TM})

$E_{\text{TM}} = \{\langle M \rangle \mid L(M) = \varnothing\}$ — описания машин с пустым языком.

Построение $f(\langle M \rangle, w) = \langle M' \rangle$: M' игнорирует свой вход, симулирует M на w и принимает при остановке.

M остановилась — язык Σ^*, зациклилась — язык $\varnothing$.



Разрешитель пустоты с инверсией ответа решал бы HALT — значит, E_{TM} неразрешима.

Остановка на пустом входе

Пример («HALT» на ε)

Язык $\mathrm{HALT}_\varepsilon = \{\langle M\rangle \mid M \text{ останавливается на пустом входе}\}$.

Сведение: $f(\langle M\rangle, w) = \langle M'\rangle$, где M' записывает w на ленту и дальше работает как M.

Тогда M останавливается на w тогда и только тогда, когда M' — на пустом входе.

Трюк один и тот же: вшить нужный вход в программу новой машины.

Так сведение сводит вопрос о паре к вопросу об одном описании.

Программа анализирует программу

Антивирус хочет решить: опасна ли эта программа?

Статический анализатор хочет решить: содержит ли код утечку?

Верификатор хочет решить: удовлетворяет ли программа спецификации?

Всё это — свойства языка, который распознаёт программа, и все сводятся от HALT.

Почему точные решатели здесь невозможны в принципе — теорема Райса на следующей лекции.

Верификация из первого семестра работала в разрешимых фрагментах — теперь видна цена этой оговорки.

Другие неразрешимые задачи

Пример (Семейство от HALT)

- **Тотальность**: останавливается ли M на каждом входе?
- **Эквивалентность**: верно ли $L(M_1) = L(M_2)$?
- **Регулярность**: является ли $L(M)$ регулярным?
- **Контекстная свобода**: является ли $L(M)$ контекстно-свободным?

Каждая задача — сведение от HALT со своей вспомогательной машиной.

Все спрашивают не об устройстве машины, а о свойствах её языка.

Лекция 11: что мы поняли?

- Диагональный аргумент из первого семестра переезжает в теорию вычислений без изменений: инвертированная диагональ не влезает в перечисление.
- HALT распознаваема симуляцией, но неразрешима: машина D, спрашивающая H о себе, ломает любой решатель.
- Сведение $A \underset{m}{\leq} B$ переносит неразрешимость: сводим известную неразрешимую к интересующей задаче.
- Пустота, тотальность, эквивалентность, регулярность — все неразрешимы одним и тем же ходом.
- Анализ программ — программа о программе — упирается в эту границу всегда, на любом языке.

Вопросы для самопроверки

1. Воспроизведите схему диагонализации Кантора и укажите, где в ней живёт инверсия.
2. Сформулируйте проблему остановки как язык.
3. Проведите доказательство неразрешимости: что делает D на входе $\langle D \rangle$ в обоих случаях?
4. Почему HALT всё-таки распознаваема?
5. Почему дополнение HALT не распознаваемо?
6. Определите сведение $A \underset{m}{\leq} B$ и объясните, в какую сторону его строить для доказательства неразрешимости.
7. Опишите сведение от HALT к проблеме пустоты языка.

Чтение к следующей паре

- m26 — «Проблема остановки», «Сведения».
- Флеш-опросник — в начале лекции 12.
- Контрольная 3 (машина Тьюринга, проблема остановки, сведение, теорема Райса) — на следующей неделе, на отдельной паре.