

Дискретная математика

Разрешимость и неразрешимость

Константин Чухарев

Осень 2026

Разрешимость

Мы можем видеть лишь небольшое расстояние вперёд, но можем видеть там много такого, что нужно сделать.

— Алан Тьюринг

Разрешимые и распознаваемые

Определение 1

Язык **разрешим**, если МТ всегда останавливается и отвечает да/нет.

Расознаваем (*recursively enumerable*): МТ принимает слова из L .

На остальных может заиклиться.

Разрешимость влечёт распознаваемость, а обратное неверно.

Вычислимо не значит практически решаемо.

Алгоритм может быть астрономически медленным.

Разрешимо и распознаваемо: примеры

Пример

- $\{0^n 1^n\}$ разрешим: МТ всегда останавливается.
- HALT распознаваема: универсальная МТ симулирует и принимает при остановке.

HALT распознаваема, но не разрешима.

Симуляция зациклившейся машины тоже зацикливается.

L разрешим тогда и только тогда, когда L и $\bar{L}$ распознаваемы.

Проблема остановки

Проблема остановки

Определение 2 (Проблема остановки)

Проблема остановки (*halting problem*): по кодировке МТ $\langle M \rangle$ и строке w узнать, останавливается ли M на w .

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

Наивный подход: симулировать M на w и ждать результата. Но если M не останавливается, ждать можно бесконечно.

Симуляция не решает проблему остановки: заиклившаяся машина заставляет ждать вечно.

Конкретный запрос

Пример (Конкретный запрос)

- M_{loop} — МТ, которая на любом входе сразу зацикливается.
- M_{acc} — МТ, которая на любом входе сразу принимает.

Тогда $\langle M_{\text{loop}} \rangle 0 \notin \text{HALT}$, а $\langle M_{\text{acc}} \rangle \varepsilon \in \text{HALT}$.

Тривиальные случаи решаются локально.

Но универсального алгоритма для всех пар $\langle M \rangle w$ не существует.

Неразрешимость

Теорема 1

HALT неразрешима.

Доказательство

Предположим, H разрешает HALT.

Построим D : на $\langle M \rangle$ спрашивает $H(\langle M \rangle \langle M \rangle)$. Если H принимает — заикливается, иначе — останавливается.

Если $D(\langle D \rangle)$ останавливается, то D на входе $\langle D \rangle$ заикливается, а если заикливается — останавливается. Обе возможности ведут к противоречию.

Нет программы, решающей проблему остановки.

Это верно для любого языка и любой архитектуры.

HALT и её дополнение

$\overline{\text{HALT}}$ — пары, на которых M закликивается — даже не распознаваема.

Иначе HALT была бы разрешима: разрешимость = распознаваемость и ко-распознаваемость.

Диагонализация

Один и тот же паттерн повторяется у Кантора, Тьюринга, Гёделя и Рассела.

В каждом случае строится объект, отличающийся от любого элемента перечисления.

Система, говорящая о себе, порождает неразрешимость.

Неразрешимость — это не наша неспособность придумать алгоритм: такого алгоритма не существует в принципе.

Это свойство выразительности.

Примеры диагонализации

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»	истинно, но недоказуемо
Тьюринг	D на своём коде	нет алгоритма

Сведения и Райс

Сведение

Определение 3

Сведение (*reduction*): $A \leq_m B$, если есть вычислимая $f: w \in A \leftrightarrow f(w) \in B$.

Решив B , решаем A .

Если A неразрешима и $A \leq_m B$, то B неразрешима.

Сводим известную неразрешимую к интересующей.

Направление сведения

Свести A к разрешимой B — ничего не доказать: разрешимость B ничего не говорит об A .

Сводить нужно неразрешимую к A :

- если H (неразрешимая) сводится к A , то A неразрешима.
- иначе решатель A решал бы и H .

Пример: пустота

Пример

$$E_{\text{TM}} = \{\langle M \rangle \mid L(M) = \emptyset\}.$$

Сведение: M' игнорирует вход и симулирует M на w . Если M останавливается (принимая или отвергая), M' принимает.

Тогда $L(M') = \emptyset$ тогда и только тогда, когда M не останавливается на w .

Если бы E разрешал пустоту, H разрешал бы HALT — противоречие.

Другие неразрешимые задачи

Пример (Неразрешимые задачи о МТ)

Каждая неразрешима — сведением от HALT:

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

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

Теорема Райса

Теорема 2 (Теорема Райса)

Пусть P — свойство распознаваемых языков:

- P зависит только от языка: если $L(M_1) = L(M_2)$, то $L(M_1) \in P$ тогда и только тогда, когда $L(M_2) \in P$.
- P нетривиально: есть M_1 с $L(M_1) \in P$ и есть M_2 с $L(M_2) \notin P$.

Тогда $\{\langle M \rangle \mid L(M) \text{ удовлетворяет } P\}$ неразрешим.

После Райса вопрос не «неразрешимо ли», а «тривиально ли».

Райс: свойства и границы

Пример (Свойства, закрытые Райсом)

Свойства нетривиальны и зависят только от языка:

- Пустота.
- Конечность.
- Регулярность.
- Контекстная свобода.
- $L(M) = \Sigma^*$.

Все неразрешимы.

Свойство «у машины не более 100 состояний» — о тексте машины, не о языке: Райс неприменим.

За пределами

Оракулы

Определение 4 (МТ с оракулом)

МТ с дополнительной лентой-оракулом и состояниями запроса.

Машина записывает строку на ленту-оракул и за один шаг узнаёт, принадлежит ли она языку-оракулу A .

Пример (Оракул для HALT)

Оракул для HALT не решает тотальность.

Вопрос «для всех ли x машина M останавливается на x » требует бесконечного числа запросов — по одному на каждый вход.

Оракул для HALT открывает одни задачи, но появляется новая проблема остановки для машин с оракулом — и так по лестнице вверх.

Арифметическая иерархия

Определение 5 (Уровни)

P — разрешимый предикат.

- Σ_1 : $\exists n.P(x, n)$ — распознаваемые языки.
- Π_1 : $\forall n.P(x, n)$ — ко-распознаваемые.
- $\Delta_1 = \Sigma_1 \cap \Pi_1$: разрешимые.
- Σ_2 : $\exists m.\forall n.P(x, m, n)$.
- Π_2 : $\forall m.\exists n.P(x, m, n)$.

Уровни Σ_k и Π_k — это порядок кванторов из лекции о логике.

$\forall x \exists y$ (у каждого есть мать) — Π_2 , а $\exists y \forall x$ — Σ_2 .

Уровни: примеры

Пример (Задачи на уровнях)

- HALT — Σ_1 -полна.
- Тотальность — Π_2 -полна.

$$\Delta_1 \subset \Sigma_1 \subset \Delta_2 \subset \Sigma_2 \subset \dots$$

Тотальность строго труднее проблемы остановки.

Гёдель о неполноте

Полнота из лекции о логике гласила: истинное доказуемо. Для арифметики это неверно.

Теорема 3 (Первая теорема Гёделя)

Пусть T — непротиворечивая рекурсивно аксиоматизируемая теория, содержащая достаточно арифметики (например, арифметику Пеано или арифметику Робинсона Q).

Тогда есть предложение φ , которое T не может ни доказать, ни опровергнуть.

Теорема 4 (Вторая теорема Гёделя)

Такая теория T не доказывает предложение $\text{Con}(T)$, выражающее непротиворечивость T .

Гёдель: доказательство

Доказательство

Предложение Гёделя φ строится так, что T доказывает $\varphi \leftrightarrow \neg \text{Prov}(\varphi)$, где Prov — формула, выражающая доказуемость.

Доказательство первой теоремы переносится внутрь T : из непротиворечивости T следует $\neg \text{Prov}(\varphi)$, а по эквивалентности — и φ .

Если бы T доказывало $\text{Con}(T)$, оно доказало бы и φ , что противоречит первой теореме. Значит, $\text{Con}(T)$ недоказуемо.

Пример: Гудштейн

Пример (Предложение Гудштейна)

Для любого старта последовательность Гудштейна конечна и достигает нуля.

Утверждение истинно (доказывается трансфинитной индукцией), но недоказуемо в арифметике Пеано (Кирби и Парис, 1982).

Та же диагонализация, что у Тьюринга.

Граница доказательств — та же граница, что у вычислений.

Занятый бобёр

Определение 6

Функция занятого бобра $\Sigma(n)$ — максимальное число единиц, которые останавливающаяся МТ с n состояниями над двоичным алфавитом $\{0, 1\}$ записывает на пустой ленте.

Функция $S(n)$ — максимальное число шагов такой МТ.

$S(n)$ — это кратчайший корректный таймаут для машин с n состояниями: любой универсальный таймаут обязан превышать $S(n)$.

Для достаточно больших n значение $\Sigma(n)$ недоказуемо в ZFC: есть явная машина с несколькими сотнями состояний, поведение которой ZFC не может определить.

Известные значения

Пример

Известные значения:

- $\Sigma(1) = 1$.
- $\Sigma(2) = 4$.
- $\Sigma(3) = 6$.
- $\Sigma(4) = 13$.

Машина с двумя состояниями записывает максимум 4 единицы.

Для $n = 5$ значение $\Sigma(5) = 4098$ найдено только в 2024 году.

Невычислимость и рост

Теорема 5 (Невычислимость)

$\Sigma(n)$ и $S(n)$ невычислимы: иначе по $\Sigma(n)$ можно было бы ограничить время симуляции и решить проблему остановки.

Для любой вычислимой f есть n_0 : $\Sigma(n) > f(n)$ для всех $n \geq n_0$.

$\Sigma(6)$ уже превосходит башню из 15 экспонент.

Применения

Статический анализ программ ограничен теоремой Райса.

Нельзя проверить любое нетривиальное семантическое свойство программ.

Анализ даёт приближения, а не точные ответы.

Компиляторы, линтеры, верификаторы — приближённые.

Точность и разрешимость несовместимы.

Что мы поняли?

- Разрешимость — остановка с ответом да/нет всегда. Распознаваемость допускает заикливание вне языка.
- Проблема остановки HALT неразрешима: диагональный аргумент опровергает любой решающий алгоритм.
- Сведение переносит неразрешимость на новые задачи. Теорема Райса закрывает все нетривиальные семантические свойства.
- Граница вычислений совпадает с границей доказательств: теоремы Гёделя — та же диагонализация, оракулы поднимают проблему остановки по лестнице вверх.

Вопросы для самопроверки

- Чем разрешимость отличается от распознаваемости?
- Сформулируйте проблему остановки.
- Докажите её неразрешимость.
- Что такое сведение и как им пользоваться?
- Сформулируйте теорему Райса.
- Что такое занятый бобёр?
- К какому уровню арифметической иерархии принадлежат проблема остановки и тотальность?
- Почему оракул для проблемы остановки не решает тотальность?
- Как теорема Райса ограничивает статический анализ?