

3.1 Определения

Опр Машина Тьюринга

Формальное определение машины Тьюринга - это кортеж $(\Sigma, \Gamma, Q, q_1, q_a, q_r, \delta)$, где

Σ - конечное непустое множество - входной алфавит, типично 0,1

Γ - конечное непустое множество, включающее в себя Σ , как подмножество, а также, по меньшей мере, еще пустой символ (бланк, пробел) - ленточный алфавит

Q - конечное множество, не пересекающееся с Γ - множество внутренних состояний

$q_1 \in Q$ - начальное состояние

$q_a \in Q$ - принимающее состояние

$q_r \in Q$ - отвергающее состояние

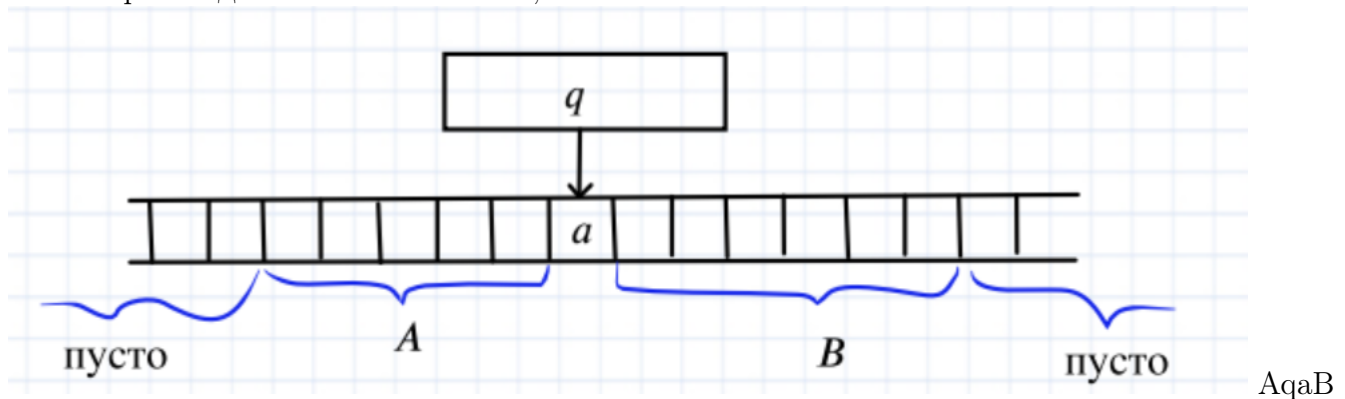
δ - функция перехода. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$

Для задач с текстовым или числовым ответом вместо q_r, q_a рассматривают одно q_0

Опр Конфигурация машины Тьюринга - данные о содержимом ленты, положении указателя и состоянии управляющего блока.

Начальная конфигурация: на ленте написан вход, машина в состоянии q , указывает на первый символ входа. У каждой конфигурации есть однозначно определяемая следующая. Если состояние завершающее, конфигурация уже не меняется.

Иначе производится замена символа, состояния и положения головки.



Вычислением на машине Тьюринга называется последовательность конфигураций, каждая из которых непосредственно следует из предыдущей по правилам этой машины.

Опр Вычислимая функция

Функция $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ называется *вычислимой*, если для некоторой машины Тьюринга выполнено:

- 1 Если $f(x)$ определена, то существует вычисление, которое начинается с qx и заканчивается $q_0f(x)$
- 2 Если $f(x)$ не определена, то не существует вычисление, которое начинается с qx и заканчивается $q_0f(x)$

Примеры

- ✓ Нигде не определённая функция вычислима (в качестве алгоритма надо взять программу, которая всегда закликивается).
- ✓ $f(x) = x$
 - $\delta(q_1, 0) = (q_0, 0, N)$

- $\delta(q_1, 1) = (q_0, 1, N)$
- $\delta(q_1,) = (q_0, , N)$

Опр Разрешимое множество

Множество $A \subset \{0, 1\}^*$ называется *разрешимым*, если для некоторой машины Тьюринга выполнено:

- 1 Если $x \in A$, то существует вычисление на этой машине, которое начинается с q_1x и заканчивается q_a
- 2 Если $x \in \bar{A}$, то существует вычисление на этой машине, которое начинается с q_1x и заканчивается q_r

Опр Перечислимое множество

Будем рассматривать машину, у которой вместо завершающих состояний есть команды печати в поток вывода: печать 0, печать пробела. Результатом работы такой машины будет конечная или бесконечная цепочка слов, разделенных пробелами.

Множество называется *перечислимым*, если существует печатающая машина, такая что:

Если $x \in A$, то x встречается в потоке вывода

Если $x \notin A$, то x не встречается в потоке вывода

Примеры

- ✓ Пустое множество является перечислимым
- ✓ Область значений/Область определения любой вычислимой функции - перечислимое множество
- × $\{n | U(n, x) \text{ определено при всех } x\}$ - неперечислимо

Опр Универсальная машина Тьюринга

Что такое? Это некоторая машина, которая получает на вход описание другой машины и вход для нее, а возвращает результат ее работы.

$$U(< M >, x) = M(x)$$

Опр Универсальная вычислимая функция.

Функция $u : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^*$ называется *универсальной вычислимой функцией*, если:

- 1 u вычислима, как функция от двух аргументов
- 2 Если $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ - вычислимая функция одного аргумента, то $\exists p \forall x u(p, x) = f(x)$

Опр Главная универсальная вычислимая функция

$U : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ - Главная Универсальная Функция, если

- 1 U вычислима
- 2 U универсальна, т.е. для любой вычислимой $f : \mathbb{N} \rightarrow \mathbb{N}$ найдется p такое, что $\forall x f(x) = U(p, x)$ (говорят, что p - это номер функции f)
- 3 U главная, т.е. для любой вычислимой $V : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ найдется всюду определенная вычислимая $s : \mathbb{N} \rightarrow \mathbb{N}$, такая что $\forall p \forall x V(p, x) = U(s(p), x)$

Интуитивный смысл: U - универсальный компилятор, V - какой-то вычислимый. Первый аргумент V - "программа", второй - "данные", s - "автоматический транслятор", переделывающие программу для V в программу для U

Опр m -сводимость

Говорят, что A m -сводится к B , если существует всюду определенная вычислимая функция $f: \mathbb{N} \rightarrow \mathbb{N}$, такая что $x \in A \iff f(x) \in B$. Обозначение: $A \leq_m B$

Опр Классы арифметической иерархии

Говорят, что множество $A \subset \mathbb{N}^k$ принадлежит классу Σ_n , если существует такое разрешимое множество $R \in \mathbb{N}^{k+1}$, что

$$(x_1, x_2, \dots, x_k) \in A \iff \exists y_1 \forall y_2 \exists y_3 \dots Q y_n [(x_1, \dots, x_k, y_1, \dots, y_k) \in R]$$

Аналогично, говорят, что $A \subset \mathbb{N}^k$ принадлежит классу Π_n , если существует такое разрешимое множество $R \in \mathbb{N}^{k+1}$, что

$$(x_1, x_2, \dots, x_k) \in A \iff \forall y_1 \exists y_2 \forall y_3 \dots Q y_n [(x_1, \dots, x_k, y_1, \dots, y_k) \in R]$$

Согласно этому определению, $\Sigma_0 = \Pi_0$ (классы Σ_0 и Π_0 совпадают с классом всех разрешимых множеств)

Σ_1 - перечислимые, Π_1 - коперечислимые

▲ S перечисливо \iff для некоторого разрешимого R верно $(x \in S \iff \exists y (x, y) \in R)$, Q коперечисливо \iff для некоторого разрешимого R верно $(x \in S \iff \forall y (x, y) \in R)$

Примеры

1 T - множество всюду определенных функций

$$p \in T \iff \forall n \exists k (U(p, n) \text{ останавливается за } k \text{ шагов}) (*)$$

(*) - разрешимое свойство $\implies T \in \Pi_2$

2 FD - множество функций с конечной областью определения

$$p \in FD \iff \exists N \forall n \forall k (n > NU(p, n) \text{ останавливается за } k \text{ шагов}) (*)$$

(*) - разрешимое свойство $\implies FD \in \Sigma_2$

Опр λ -термы

λ -терм строится по индукции

- 1 Переменная является λ -термом
- 2 (Операция аппликации): Если M и N суть лямбда-термы, то (MN) - тоже лямбда-терм.
Смысл: в функцию M вместо переменное подставляем N
- 3 (Операция λ -абстракции): Если M - терм, а x - переменная, то $(\lambda x.M)$ - тоже терм
Смысл: выражение M теперь рассматривается как функция от x

Опр α -конверсия

α -конверсия - это замена связанной переменной. $\lambda x.M \rightarrow \lambda z.M(z/x)$

Пример

✓ $\lambda x.xy \rightarrow \lambda z.zy$ - так можно

✓ $\lambda x.xy(\lambda x.xy) \rightarrow \lambda z.zy(\lambda x.xy)$ - и так можно

× $\lambda x.xy \rightarrow \lambda y.y$ - а вот так нельзя

× $\lambda x.x(\lambda y.xy) \rightarrow \lambda y.y(\lambda y.yy)$ - и так тоже нельзя! Тут переменная, полученная после замены, попала под воздействие уже имеющегося квантора

Опр β -редукция

Замена аргумента функции на какое-то значение. $(\lambda x.M)N \rightarrow M(N/x)$

Пример

✓ $\sin x$ при $x = 2$ равен $\sin 2$

× $(\lambda x.(x\lambda y.xy))y \rightarrow y\lambda y.yy$ - так нельзя

Опр Нормальная форма

Говорят, что терм M находится в *нормальной форме*, если к нему нельзя применить β -редукцию даже после нескольких α -конверсий

Говорят, что N - нормальная форма M , если $M = N$ и N находится в нормальной форме.

Не у всех термов есть нормальная форма.

Пример

$\Omega = (\lambda x.xx)(\lambda x.xx)$

Опр Нумералы Чёрча

Семантика нумералов Черча.

Формально, число k соответствует преобразованию функции f в ее k -ую итерацию

$$\begin{aligned}\bar{0} &= fx.x \\ \bar{1} &= fx.fx \\ \bar{2} &= fx.f(fx) \\ &\vdots \\ \bar{n} &= \lambda fx.f(f \dots f(fx) \dots) - n \text{ раз } f\end{aligned}$$

Опр Комбинатор

Комбинатором называется замкнутый λ -терм (без свободных переменных)

Говорят, что комбинатор G представляет функцию $g : \mathbb{N}^k \rightarrow \mathbb{N}$, если для любых n_1, \dots, n_k выполнено:

$$G\bar{n}_1\bar{n}_2\dots\bar{n}_k = \overline{g(n_1, \dots, n_k)}.$$

Если g не определена, то у $G\bar{n}_1\bar{n}_2\dots\bar{n}_k$ нет нормальной формы)

Пример

1 Inc - прибавление 1. $\text{Inc } \bar{n} = \bar{n} + 1 :$

$$\text{Inc} = \lambda nfx.f(nfx)$$

2 Add - сложение. $\text{Add } \bar{n}\bar{m} = \overline{n + m} :$

$$\text{Add} = \lambda mnfx.mf(nfx)$$

3 Mult - умножение:

$$\text{Mult} = \lambda mnfx.m(nf)x$$

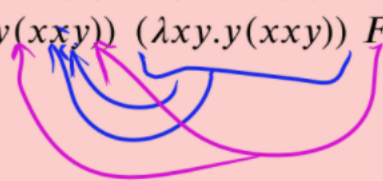
4 Exp - возведение в степень:

$$\text{Exp} = \lambda mnfx.nmfx$$

Опр Комбинатор неподвижной точки

Y - комбинатор неподвижной точки, если для любого F верно $YF = F(YF)$

Пример

$$\begin{aligned} \text{Пример : } Y &= (\lambda x y. y(xxy))(\lambda x y. y(xxy)) \\ YF &= (\lambda x y. y(xxy)) (\lambda x y. y(xxy)) F \\ &= F((\lambda x y. y(xxy))(\lambda x y. y(xxy)) F) = F(YF) \end{aligned}$$


3.2 Простые утверждения

3.2.1 Композиция вычислимых функций вычислима

Пусть машина M_f вычисляет функцию f , а машина M_g — функцию g . Тогда функцию $f(g(x))$ можно вычислить машиной M_g , которая вместо своего конечного состояния переходит в начальное состояние машины M_f (при этом для самой машины M_f нужны новые состояния, не пересекающиеся с состояниями M_g).

3.2.2 Существование невычислимых функций, неразрешимых и неречислимых множеств

Невычислимая

Функции, о которых идет речь, представляют собой функции, заданные и принимающие значения в множестве слов в алфавите A . Ясно, что множество слов в алфавите A счетно. Следовательно, рассматривается множество всех функций, заданных на счетном множестве и принимающих значения в счетном же множестве. Как известно, это множество имеет мощность континуума. С другой стороны, поскольку множество всевозможных машин Тьюринга счетно, то множество функций, вычислимых по Тьюрингу, также счетно. Континуальная мощность строго больше счетной. Следовательно, существуют функции, не вычислимые по Тьюрингу.

Неразрешимое и неречислимое множество

Алгоритмов (и поэтому разрешимых/перечислимых подмножеств натурального ряда) счётное число, а всех подмножеств натурального ряда несчётное число. Значит, из соображения мощности найдутся неразрешимые и неречислимые множества

3.2.3 Перечислимость любого разрешимого множества

По определению разрешимого множества, существует такая машина, что если $x \in A$, то существует вычисление, начинающееся в q_1x и заканчивающееся в q_a . Это значит, что для этой машины все $x \in A$ встречаются в потоке вывода. Значит, множество A перечислимо

3.2.4 Разрешимость любого конечного множества.

Алгоритм разрешения любого конечного множества S содержит таблицу элементов множества S , вход сравнивается по очереди со всеми элементами таблицы. В случае совпадения выдаем 1, иначе 0

3.2.5 Замкнутость классов разрешимых и перечислимых множеств относительно пересечения и объединения, класса разрешимых относительно дополнения

Пересечение и объединение перечислимых множеств - перечислимое множество

Если X и Y перечисляются алгоритмами A и B , то их объединение перечисляется алгоритмом, который параллельно выполняет по шагам A и B и печатает всё, что печатают A и B . С пересечением немного сложнее — результаты работы A и B надо накапливать и сверять друг с другом; что появится общего — печатать.

Пересечение, объединение, дополнение разрешимых множеств - разрешимое множество

Пересечение, объединение, дополнение - это просто композиция соответствующей характеристической функции и булевой функции.

- Для дополнения достаточно рассмотреть тот же алгоритм, что и для разрешения множества A . Вместо единицы печатать 0, вместо 0 - единицу.
- $\chi_{A \cup B}(x) = \chi_A \vee \chi_B$ - вычислима
- $\chi_{A \cap B}(x) = \chi_A \wedge \chi_B$ - вычислима

3.2.6 Существование вычислимой в обе стороны биекции между \mathbb{N}^2 и \mathbb{N}

$(x, y) \mapsto (2x + 1)2^y$

3.2.7 Подмножество разрешимого (перечислимого) множества не обязательно разрешимо (перечислимо), и наоборот

Подмножество разрешимого/перечислимого множества может быть неразрешимо/-неперечислимо

Любое множество, в т.ч. неразрешимое/неперечислимое, является подмножеством \mathbb{N} , которое разрешимо/перечислимо

Подмножество неразрешимого/неперечислимого множества может быть разрешимо/перечислимо Достаточно в любом множестве выбрать конечное подмножество, тогда оно разрешимо/перечислимо

3.2.8 Свойства m -сводимости: транзитивность, сводимость дополнений, разрешимость множества, m -сводимого к разрешимому, перечислимость множества, m -сводимого к перечислимому

Основные свойства m -сводимости :	
0) Рефлексивность $A \leq_m A$.	
1) Транзитивность : $A \leq_m B, B \leq_m C \Rightarrow A \leq_m C$.	
Это следует из того, что композиция вычислимых функций вычислима	
$x \in A \Leftrightarrow f(x) \in B \Leftrightarrow g(f(x)) \in C$	
2) Сводимость к разрешимому : $A \leq_m B, B$ разрешимо $\Rightarrow A$ разрешимо	
$x \in A \Leftrightarrow f(x) \in B \Leftrightarrow R(f(x)) = 1$	
$R \circ f$ вычислимо и будет программой, разрешающей A	
3) Сводимость к перечислимому : $A \leq_m B, B$ перечислимо $\Rightarrow A$ перечислимо	
Например, как выше, но в качестве R нужно взять программу, вычисляющую полухарактеристическую функцию	
4) Сводимость дополнений : $A \leq_m B \Leftrightarrow \bar{A} \leq_m \bar{B}$	
$x \in \bar{A} \Leftrightarrow \neg(x \in A) \Leftrightarrow \neg(f(x) \in B) \Leftrightarrow f(x) \in \bar{B}$ (т.е. годится та же самая функция)	
5) сводимость разрешимых множеств	
Если A разрешимо, а B и \bar{B} непусты, то $A \leq_m B$	
Если есть $b_0 \in B$ и $b_1 \in \bar{B}$, то рассмотрим $f(x) = \begin{cases} b_0, & x \in A \\ b_1, & x \in \bar{A} \end{cases}$	

3.2.9 Вложенность классов в арифметической иерархии

$$\Sigma_k \subset \Sigma_{k+1}, \Sigma_k \subset \Pi_{k+1}, \Pi_k \subset \Sigma_{k+1}, \Pi_k \subset \Pi_{k+1}$$

▲ Добавляем нужный квантор по фиктивной переменной. Например, $\exists y(x, y) \in R \iff \exists y \forall z(x, y, z) \in R \times \mathbb{N} \iff \forall t \exists y(x, y, t) \in R \times \mathbb{N}$ (из Σ_1 в Π_2, Σ_2) ■

3.2.10 Замкнутость классов арифметической иерархии относительно объединения и пересечения

Пусть $A, B \in \Sigma_k$

$$\begin{aligned} x \in A &\iff \exists y_1 \forall y_2 \dots \exists y_k(x, y_1, \dots, y_k) \in R \\ x \in B &\iff \exists z_1 \forall z_2 \dots \exists z_k(x, z_1, \dots, z_k) \in Q \\ x \in A \cap B &\iff (\exists y_1 \forall y_2 \dots \exists y_k(x, y_1, \dots, y_k) \in R) \vee (\exists z_1 \forall z_2 \dots \exists z_k(x, z_1, \dots, z_k) \in Q) \iff \\ &\iff \exists(y_1, z_1) \forall(y_2, z_2) \dots \exists(y_k, z_k)((x, y_1, \dots, y_k) \in R \wedge (x, z_1, \dots, z_k) \in Q) \end{aligned}$$

что является разрешимым свойством следующего кортежа: $((x, (y_1, z_1), \dots, (y_k, z_k)))$.

Значит, $A \cap B \in \Sigma_k$. Для объединения аналогично

3.2.11 Пример λ -терма, к которому можно применить β -редукцию только после α -конверсии

$$(\lambda xy.x)y \xRightarrow{\alpha} (\lambda xt.x)y \xRightarrow{\beta} \lambda t.y$$

3.2.12 Пример λ -терма, не имеющего нормальной формы

$$(\lambda x.xx)(\lambda x.xx)$$

3.2.13 Построение комбинаторов сложения и умножения для нумералов Чёрча (с доказательством корректности)

Add - сложение.

$$\begin{aligned} \text{Add } \overline{m}\overline{n} &= (\lambda mnfx.mf(nfx))\overline{m}\overline{n} = (\lambda nfx.\overline{m}f(nfx))\overline{n} = (\lambda nfx.(\lambda gy.\underbrace{g(g\dots(gy))}_m))f(nfx))\overline{n} = \\ &= (\lambda nfx.(\lambda y.\underbrace{f(f\dots(fy))}_m))(nfx))\overline{n} = \lambda fx.(\lambda y.\underbrace{f(f\dots(fy))}_m)(\overline{n}fx) = \\ &= \lambda fx.(\lambda y.\underbrace{f(f\dots(fy))}_m)(\lambda gt.\underbrace{g(g\dots(gt))}_n)fx) = \lambda fx.(\lambda y.\underbrace{f(f\dots(fy))}_m)(\underbrace{f(f\dots(fx))}_n)) = \\ &= \lambda fx.(\underbrace{f(f\dots(fx))}_{m+n}) = \overline{m} + \overline{n} \end{aligned}$$

Mult - умножение:

$$\begin{aligned} \text{Mult } \overline{m}\overline{n} &= (\lambda mnfx.m(nfx))\overline{m}\overline{n} = (\lambda nfx.\overline{m}(nfx))\overline{n} = (\lambda nfx.(\lambda gy.\underbrace{g(g\dots(gy))}_m))(nfx))\overline{n} \\ &= \lambda fx.(\lambda gy.\underbrace{g(g\dots(gy))}_m)(\overline{n}fx) = \lambda fx.(\lambda y.\underbrace{\overline{n}f(\overline{n}f\dots(\overline{n}fy))}_m)x = \\ &= \lambda fx.(\lambda y.\underbrace{\overline{n}f(\overline{n}f\dots(\overline{n}f(\lambda gt_1.\underbrace{g(g\dots(gt_1))}_n)f y))}_n)x \\ &= \lambda fx.(\lambda y.\underbrace{\overline{n}f(\overline{n}f\dots(\overline{n}f(\underbrace{f(f\dots(fy))}_m))}_n))x = \lambda fx.(\lambda y.\underbrace{\overline{n}f(\overline{n}f\dots(\overline{n}f(\underbrace{f(f\dots(fy))}_m))}_n))x = \dots = \\ &= \lambda fx.(\lambda y.\underbrace{f(f\dots(fy))}_{n+m-1}))x = \lambda fx.\underbrace{f(f\dots(fx))}_{n*m} = \overline{m} * \overline{n} \end{aligned}$$