3.1 Определения

Опр *Машина Тьюринга*
Формальное определние 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

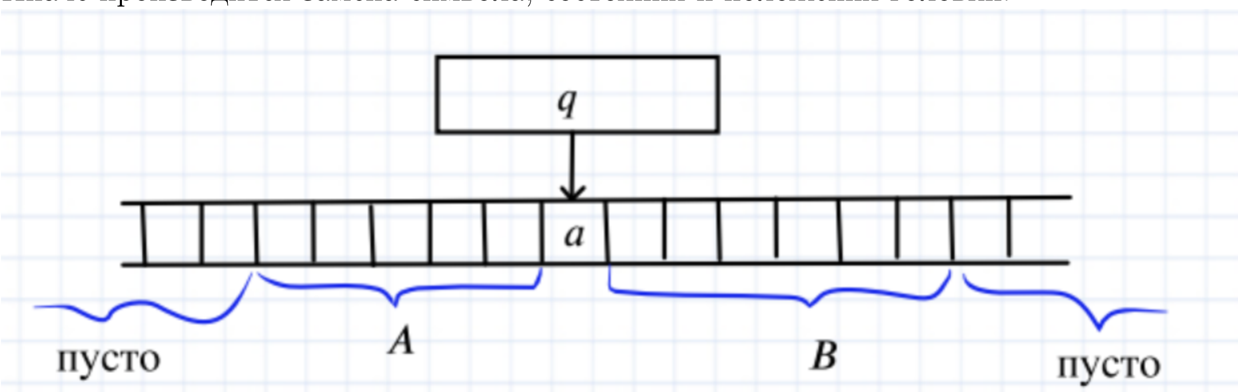
q_1 $\in Q$ - начальное состояние

q_a $\in Q$ - принимающее состояние

q_r $\in Q$ - отвергающее состояние

δ - функция перехода. $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$, где L - перемещение влево, R - вправо, N - никуда

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 упавляющего блока.
Начальная конфигурация: на ленте написан вход, машина в состоянии q, указывает на первый символ входа. У каждой конфигурации есть однозначно опредляемая следующая. Если состояние завершающее, конфигурация уже не меняется.
Иначе производится замена символа, состояния и положения головки.



AqaB

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

Пример смены конфигураций

Если, например, $B = bB'$, и $\delta(q,a) = (s,c,R)$, то следующая конфигурация за $AqaB$ будет $AcsbB'$
Если $B = \varepsilon$ и $\delta(q,a) = (s,c,R)$, то следующая конфигурация будет $Acs\#$ (# — бланк)

q,s - состояния
Опр *Вычислимая функция*
Функция $f : \{0,1\}^* \to \{0,1\}^*$ называется *вычислимой*, если для некоторой машины Тьюринга выполнено:

1 Если $f(x)$ определена, то существует вычисление, которое начинается с qx и заканчивается $q_0 f(x)$

2 Если $f(x)$ не определена, то не существует вычисление, которое начинается с qx и заканчивается $q_0 f(x)$

Примеры

✓ Нигде не определённая функция вычислима (в качестве алгоритма надо взять программу, которая всегда зацикливается).

✓ Пусть $\Sigma = \{0, 1\}$. Опишите машины, вычисляющие функции: f(x) = x, f(x) = 0, нигде не определённую.

- $\delta(q_1, 0) = (q_0, 0, N)$
- $\delta(q_1, 1) = (q_0, 1, N)$
- $\delta(q_1, \#) = (q_0, \#, N)$

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 $x \in \overline{A}$, 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 бесконечноая 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

× $\{n|U(n, x)$ 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 $u : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ называется *универсальной вычислимой функцией*, если:

1 u вычислима, как функция от двух аргументов

2 Если $f : \{0,1\}^* \to \{0,1\}^*$ - вычислимая функция одного аргумента, то $\exists p \forall x \; u(p, x) = f(x)$

Опр *Главная универсальная вычислимая функция*
$U : N \times N \to N$ - Главная Универсальная Функция, если

1 U вычислима

2. U универсальна, т.е. для любой вычислимой $f : \mathbb{N} \to \mathbb{N}$ 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 $V : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ найдется всюду определенная вычислимая $s : \mathbb{N} \to \mathbb{N}$, 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 трасплятор¨ , переделывающие программу для V в программу для U

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

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

$$x \in A \iff \exists y_1\ \forall y_2\ \exists y_3 .... \mathcal{Q} y_n [(x, y_1, ..., y_k) \in R]$$

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

$$x \in A \iff \forall y_1\ \exists y_2\ \forall y_3 .... \mathcal{Q} y_n [(x, y_1, ..., y_k) \in R]$$

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 перечислимо $\iff$ для некоторого разрешимого R верно ($x \in S \iff \exists y\ (x, y) \in R$), Q коперечислимо $\iff$ для некоторого разрешимого R верно ($x \in S \iff \forall y\ (x, y) \in R$)
Примеры

1. T - множество всюду определенных функций
   $p \in T \iff \forall n \exists k (U(p, n)$ останавливается за k шагов)(*)
   (*) - разрешимое свойство $\implies T \in \Pi_2$

2. FD - множество функций с конечной областью определения
   $p \in FD \iff \exists N \forall n\ \forall k (n > N U(p, n)$ останавливается за k 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 связаной переменной. $\lambda x.M \to \lambda z.M(z/x)$
Ограничение: не должно возникать конфликтов имен. Переменные из M не должны попадать под действие λ-квантора

Пример

- ✓ $\lambda x.xy \to \lambda z.zy$ - так можно
- ✓ $\lambda x.xy(\lambda x.xy) \to \lambda z.zy(\lambda x.xy)$ - и так можно
- × $\lambda x.xy \to \lambda y.yy$ - а вот так нельзя
- × $\lambda x.x(\lambda y.xy) \to \lambda y.y(\lambda y.yy)$ - и так тоже нельзя! Тут переменная, полученнная после замены, попала под воздействие уже имеющегося квантора

Опр *β-редукция*
Замена аргумента функции на какое-то значение. $(\lambda x.M)N \to M(N/x)$
Ограничение: не должно возникать конфликтов имен. В N не должно быть переменных, по которым стоят λ-кванторы в M

Пример

- ✓ sin x при x = 2 равен sin 2
- × $(\lambda x.(x\lambda y.xy))y \to y\lambda y.yy$ - 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

$$\overline{0} = fx.x$$
$$\overline{1} = fx.fx$$
$$\overline{2} = fx.f(fx)$$
$$\dots$$
$$\overline{n} = \lambda fx.f(f.....f(fx))...) \text{ - n раз f}$$

Опр *Комбинатор*
Комбинатором называется замкнутый λ-терм (без свободных переменных)
Говорят, что комбинатор G представляет функцию $g : \mathbb{N}^k \to \mathbb{N}$, если для любых $n_1, ..., n_k$ выполнено:

$G\overline{n_1}\overline{n_2}...\overline{n_k} = \overline{g(n_1, ..., n_k)}$. Если g не определена, то у $G\overline{n_1}\overline{n_2}...\overline{n_k}$ нет нормальной формы)

Пример

1. Inc - прибавление 1. Inc $\overline{n} = \overline{n} + 1$:
   $Inc = \lambda nfx.f(nfx)$
2. Add - сложение. Add $\overline{nm} = \overline{n+m}$:
   $Add = \lambda mnfx.mf(nfx)$
3. Mult - умножение:
   $Mult = \lambda mnfx.m(nf)x$
4. Exp - возведение в степень:
   $Exp = \lambda mnfx.nmfx$

Опр *Комбинатор неподвижной точки*
Y - комбинатор неподвижнй точки, если для любого F верно YF = F(YF)

Пример

Пример : $Y = (\lambda xy.y(xxy))(\lambda xy.y(xxy))$

$$YF = (\lambda xy.y(xxy))\ (\lambda xy.y(xxy))\ F$$

$$= F((\lambda xy.y(xxy))(\lambda xy.y(xxy))F) = F(YF)$$

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_f).

3.2.2 Существование невычислимых функций, неразрешимых и неперечислимых 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 неперечислимое 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 неперечислимые 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 я 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 $\mathbb{N}$

$(x, y) \mapsto (2x + 1)2^y$
Это биекция - очевидно, так как любое число представимо таким образом,причем разным числам соответствуют разные представления, и для любого числа вида $(2x + 1)2^y$ найдется число из N.
Опишем алгоритм вычисления

=> Очевидно, такая функция вычислима, как функция от двух аргументов

<= Делим число на 2, пока оно четно. Получаем отсюда y. Потом вычитаем один, делим на 2, получаем x.

Таким образом, построенная биекция вычислима в обе стороны, а значит, подходит под условие

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 $A \le_m A$.

1) Транзитивность : $A \le_m B,\ B \le_m C \Rightarrow A \le_m C$.

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 : $A \le_m B$, 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 : $A \le_m B$, B перечислимо $\Rightarrow$ A перечислимо

Например, как выше, но в качестве R нужно взять программу, вычисляющую полухарактеристическую функцию

4) Сводимость дополнений : $A \le_m B \Leftrightarrow \overline{A} \le_m \overline{B}$

$x \in \overline{A} \Leftrightarrow \neg(x \in A) \Leftrightarrow \neg(f(x) \in B) \Leftrightarrow f(x) \in \overline{B}$ (т.е. годится та же самая функция)

5) сводимость разрешимых множеств

Если A разрешимо, а B и $\overline{B}$ непусты, то $A \le_m B$

Если есть $b_0 \in B$ и $b_1 \in \overline{B}$, то рассмотрим $f(x) = \begin{cases} b_0, x \in A \\ b_1, x \in \overline{A} \end{cases}$

3.2.9 Вложенность классов в арифметической иерархии

$\Sigma_k \subset \Sigma_{k+1}, \Sigma_k \subset \Pi_{k+1}, \Pi_k \subset \Sigma_{k+1}, \Pi_k \subset_{k+1}$
▲ 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 $A, B \in \Sigma_k$

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

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

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

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.
Add $\overline{mn}$= $(\lambda mnfx.mf(nfx))\overline{mn}$ = $(\lambda nfx.\overline{m}f(nfx))\overline{n}$ = $(\lambda nfx.(\lambda gy.\underbrace{g(g...(g}_{m}y)...))f(nfx))\overline{n}$ = $(\lambda nfx.(\lambda y.\underbrace{f(f...(f}_{m}y)...))(nfx))\overline{n} = \lambda fx.(\lambda y.\underbrace{f(f...(f}_{m}y)...))(\overline{n}fx) =$

$\lambda fx.(\lambda y.\underbrace{f(f...(f}_{m}y)...))(\lambda gt.\underbrace{g(g...(g}_{n}t)...))fx) = \lambda fx.(\lambda y.\underbrace{f(f...(f}_{m}y)...))(\underbrace{f(f...(f}_{n}x)...))) =$
$\lambda fx.(\underbrace{f(f...(f}_{m+n}x)...)) = \overline{m} + \overline{n}$

Mult - умножение:
$Mult\ \overline{mn} = (\lambda mnfx.m(nf)x)\overline{mn} = (\lambda nfx.\overline{m}(nf)x)\overline{n} = (\lambda nfx.(\lambda gy.\underbrace{g(g...(g}_{m}y)...))(nf)x)\overline{n}$

$= \lambda fx.(\lambda gy.\underbrace{g(g...(g}_{m}y)...))(\overline{n}f)x = \lambda fx.(\lambda y.\underbrace{\overline{n}f(\overline{n}f...(\overline{n}f}_{m}y)...))x =$

$\lambda fx.(\lambda y.\underbrace{\overline{n}f(\overline{n}f...\overline{n}f(\lambda gt_1.\underbrace{g(g...(g}_{n}t_1)...))f}_{m}y)...))x$

$= \lambda fx.(\lambda y.\underbrace{\overline{n}f(\overline{n}f...\overline{n}f}_{m-1}(\underbrace{f(f...(f}_{n}y)...)))...))x = \lambda fx.(\lambda y.\underbrace{\overline{n}f(\overline{n}f...\overline{n}f}_{m-2}(\underbrace{f(f...(f}_{n+n}y)...)))...))x = ... =$
$\lambda fx.(\lambda y.(\underbrace{f(f...(f}_{n*m}y)...)))...))x = \lambda fx.\underbrace{f(f...(f}_{n*m}x)..) = \overline{m} * \overline{n}$