Теоретическая информатика.

Лектор — А.С. Охотин, Э.А. Гирш и Д.О. Соколов
Создатель конспекта — Глеб Минаев *

TODOs

Содержание

Литература:

- ...

Страницы курса:

- Часть курса, прочитанная А.С. Охотиным.

1 Вычислимость

Определение 1. *Машина Тьюринга* — это реализация понятия вычисления. Она заключается в том, что имеется бесконечная в обе стороны клетчатая лента с записанным на ней конечным словом (по букве на клетку) — входные данные вычисления — и головка — вычисляющий аппарат, которая в каждый момент времени смотрит на какую-то клетку ленты и имеет в себе некоторое внутреннее состояние вычислений. Каждым своим действием она читает символ в клетке, на которую смотрит, и в зависимости от прочитанного символа и внутреннего состояния в данный момент она пишет в клетку, на которую смотрит новый символ, стирая старый, переходит на новую клетку и меняет внутренне состояние.

Формально машина Тьюринга M — это совокупность

- входного алфавита Σ — (конечного) алфавита входных данных,
- рабочего алфавита Γ — (конечного) алфавита, над которым мы работаем, что $\Gamma \supseteq \Sigma \sqcup \{\textvisiblespace\}$,
- конечного множества (внутренних) состояний Q,
- начального состояния $q_0 \in Q$,
- функции переходов $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1; +1\}$
- и дополнительных объектов и условий в зависимости от предназначения машины.

*Оригинал конспекта расположен на GitHub. Также на GitHub доступен репозиторий с другими конспектами.

Входными данными программы является конечная строка $s_1 \dots s_l$, которая превращается в запись $\{a_i\}_{i\in\mathbb{Z}}$ на ленте по правилу

$$a_i = \begin{cases} s_{i+1} & \text{если } 0 \leqslant i \leqslant l-1, \\ \text{␣} & \text{иначе.} \end{cases}$$

Состояние каждого момент вычисления — это совокупность

- состояния ленты $\{a_i\}_{i\in\mathbb{Z}}$, где $a_i \in \Sigma$ и все a_i кроме конечного числа элементов являются пробелом ␣,
- положения головки $n \in \mathbb{Z}$ на ленте и
- состояния головки $q \in Q$.

Состояние начального момента вычислений образуется из

- состояния ленты, полученного из входных данных,
- положения головки по умолчанию $n = 0$
- начального состояния головки q_0.

Каждым шагом состояние $(\{a_i\}_{i\in\mathbb{Z}}, n, q)$ заменяется на состояние $(\{a'_i\}_{i\in\mathbb{Z}}, n', q')$, где $a'_i = a_i$ для всех $i \in \mathbb{Z} \setminus \{n\}$ и

$$(q', a'_n, n' - n) := \delta(q, a_n).$$

Далее есть два вида предназначений машины:

1. **Распознование свойства.** В этом случае мы хотим уметь распознавать разные конечные строки, т.е. чтобы машина отвечала “да” или “нет” на вопрос “Лежит ли эта строка в заданном семействе?”. В таком случае у машины выделяются *принимающее состояние* $q_{acc} \in Q$ и *отвергающее состояние* $q_{rej} \in Q$, и когда машина переходит в одно из этих двух состояний, вычисления прекращаются. В таком случае $\delta(q, a)$ должно быть не определено в случае $q \in \{q_{acc}; q_{rej}\}$.
2. **Вычисление функции.** В этом случае мы хотим вычислять значение некоторой функции $M : \Sigma^* \to \Sigma^*$, т.е. чтобы машаина после некоторого количества действий говорила “Готово.” и оставляла на ленте конечное слово в том же формате, в котором производится ввод. В таком случае у машины выделяется *состояние останова* $q_{halt} \in Q$, и когда машина переходит в это состояние, вычисления прекращаются, а на ленте должно остаться слово в правильном формате. В таком случае $\delta(q, a)$ должно быть не определено в случае $q = q_{halt}$.

Замечание. Фактически множество результатов работы машины состоит из переходов в одно из терминальных состояний (q_{acc}, q_{rej} или q_{halt}) и попадание в “вечный цикл”.

Теорема 1. *В определении машина Тьюринга множество возможных смещений головки $\{+1; -1\}$ можно сменить на некоторое $S \subseteq \mathbb{Z}$. Тогда если S конечно и всякое целое число представляется в виде суммы хотя бы одного значения (возможно, с повторами) из S, то определение получается равносильным.*

Доказательство. Действительно, пусть у нас имеется машина с множеством возможных сдвигов S_1, а у нас есть множество возможных сдвигов S_2, что всякое число из S_1 представляется в виде суммы некоторых (хотя бы одного и, возможно, с повторениями) членов из S_2. Тогда для всякой перехода между состояниями $q_1 \to q_2$ можно создать такие фальшсостояния, что смещение на значение из S_1 будет заменено на несколько смещений на значения из S_2 с тем же итогом. Таким образом мы можем всякую машину на S_1 поменять на машину на S_2 и большим числом состояний.

Таким образом множества смещений S_1 и S_2 реализуют равносильные модели, если всякое значение из S_1 расскладывается в сумму значений S_2 и наоборот. При для $S_1 = \{+1; -1\}$ переход $S_2 \to S_1$ очевиден, а переход $S_1 \to S_2$ означает, что 1 и -1 раскладываются в суммы некоторых чисел из S_2, что равносильно разложимости всякого целого. □

Следствие 1.1. *Вместо* $\{+1; -1\}$ *для удобства можно также подразумевать* $\{+1; 0; -1\}$.

Замечание. В данный момент мы будем рассматривать только задачу распознавания свойства.

Определение 2. *Язык машины* M или *язык, распознаваемый машиной* M, — это множество $L(M) \subseteq \Sigma^*$ входных строк, которые машина принимает. Т.е. это множество всех конечных строк $w \in \Sigma^*$ таких, что машина M на входе w завершает работу и принимает данный вход.

Язык, распознаваемый какой-нибудь машиной, называется *рекурсивно-перечислимым*.

Язык, распознаваемый какой-нибудь машиной, которая останавливается на любом входе, называется *рекурсивным*.

Пример 1. Пусть $\Sigma = \{a; b\}$, $L = \{a^n b^n\}_{n \geqslant 0}$. Тогда L распознаётся машиной Тьюринга (которая причём останавливается на всяком входе!).

Действительно, давайте напишем машину, которая будить ездить из одного конца нашей строки в другую и стирать a справа и b слева, и, если получит пустую строку, примет вход, а если поймет, что в какой-то момент получается что-то неправильное, то отвергнет вход. Формально, возьмём $Q = \{q_0; q_1; q_2; q_3; q_{acc}; q_{rej}\}$ и $\Gamma = \{a; b; \text{␣}\}$, а функцию перехода опишем следующей таблицей:

$Q \backslash \Gamma$	a	b	␣
q_0	q_1, ␣, $+1$	q_{rej}	q_{acc}
q_1	q_1, a, $+1$	q_1, b, $+1$	q_2, ␣, -1
q_2	q_{rej}	q_3, ␣, -1	q_{rej}
q_3	q_3, a, -1	q_3, b, -1	q_0, ␣, $+1$

Т.е. во с помощью q_0 и q_2 мы стираем a слева и b справа соответственно, а с помощью q_1 и q_2 перемещаемя до конца вправо и влево соответственно.

Теорема 2. *Есть язык, нераспознаваемый никакой машиной Тьюринга.*

Доказательство. Несложно видеть, что для всякого конечного непустого алфавита Σ количество $|\Sigma^*|$ конечных строк над Σ счётно, а значит языков над Σ континуум. При этом машин Тьюринга с входным алфавитом Σ счётное число. Значит почти все языки не распознаются машинами Тьюринга.

При этом есть довольно простые конкретные примеры нераспознаваемых языков. □

Определение 3. *Запись* $\sigma(M)$ *машины Тьюринга* M — это конечной битовая строка, созданная по следующему правилу. Пусть у машины $M = (\Sigma, \Gamma, Q, q_0, \delta, q_{acc}, q_{rej})$

- $\Sigma = \{a_1; \ldots; a_l\}$,

- $\Gamma = \Sigma \cup \{a_{l+1}; \ldots; a_m\}$, где причём $a_m = \text{␣}$,
- $Q = \{q_0; \ldots; q_{n-1}\}$, где причём $q_{acc} = q_{n-2}$, $q_{rej} = q_{n-1}$.

Тогда запишем всё в унарной системе счисления, т.е. (всякий абстрактный) массив числовых данных будет храниться в виде последовательностей единиц, длины которых будут равняться соответствующим значениям массиве, разделённых нулями. Тогда машина M будет записана строкой

$$\sigma(M) := 1^l\,0\,1^m\,0\,1^n\,0 \prod_{\delta(q_i,a_j)=q_{i'},a_{j'},d} 1^i\,0\,1^j\,0\,1^{i'}\,0\,1^{j'}\,0\,1^{d+1}$$

(где умножение, безусловно, — конкатенация, а $\prod$ — конкатенация нескольких строк).

Определение 4. Языки L_0 и L_1 — это

$$L_0 := \{\sigma(M) \mid \sigma(M) \notin L(M)\} \qquad \text{и} \qquad L_1 := \{\sigma(M) \mid \sigma(M) \in L(M)\}.$$

Теорема 3.

1. *L_0 не рекурсивно-перечислимый.*
2. *L_1 рекурсивно-перечислимый.*
3. *L_1 рекурсивный.*

Доказательство.

1. Предположим противное, т.е. есть машина M, распознающая L_0. Тогда если M принимает $\sigma(M)$, то $\sigma(M) \in L(M)$, но тогда $\sigma(M) \notin L$. Если же M не принимает $\sigma(M)$, то $\sigma(M) \notin L(M)$, но тогда $\sigma(M) \in L$. Противоречие наподобие парадокса Рассела. Значит нет никакой машины, распознающей L_0.

2. Неформально опишем машину, которая будет распознавать L_1. Идея машины заключается в том, что получая на вход $\sigma(M)$, после этого запсанного кода машины M она запишет код начального состояния q_0 машины M, а затем код $\sigma(M)$ как входную строчку для M. После этого она начнёт моделировать работу машины M на входе $\sigma(M)$. Состояние головки будет писаться перед клеткой, на которую головка смотрит. Если места слева будет не хватать, то вся запись будет просто сдвигаться вправо.

3. Покажим, что всякая машина, распознающая L_1 зацикливается при некотором входе. Предположим противное, т.е. есть машина $\widetilde{M}$, которая распознаёт L_1 и всегда остансавливается. Тогда построим $\widehat{M}$ по алгоритму:

   - Проверить правда ли, что на было подано некоторое $\sigma(M)$. Если нет, то отвергнуть.
   - Работать как $\widetilde{M}$ на $\sigma(M)$. Если $\widetilde{M}$ принимает, то отвергнуть. Если отвергает — принять.

   Тогда $\widehat{M}$ распознаёт L_0 — противоречие. Значит машины $\widetilde{M}$ не существует.

□

Теорема 4. *Язык*

$$L_\varnothing := \{\sigma(M) \mid L(M) = \varnothing\}$$

не рекурсивно-перечислимый.

Доказательство. Предположим противное, т.е. есть машина $M_\varnothing$, которая распознаёт $L_\varnothing$. Тогда построим машину M_0 по следующему алгоритму.

1. Проверить правда ли, что на было подано некоторое $\sigma(M)$. Если нет, то отвергнуть.
2. Построить $\sigma(M')$, где машина M' работает по следующему алгоритму.
   - Стереть входную строку.
   - Написать $\sigma(M)$.
   - Запустить M.
3. Запустить $M_\varnothing$ на $\sigma(M')$.

M' принимает любую строку, если M принимает $\sigma(M)$, отвергает любую строку, если M отвергает $\sigma(M)$, и зацикливается, если M зацикливается на $\sigma(M)$. Следовательно $M_\varnothing$ примет $\sigma(M')$ тогда и только тогда, когда $\sigma(M) \notin L(M)$. Таким образом M_0 распознаёт L_0. □

Определение 5. Для всякого свойства P языков можно определить язык

$$L_P := \{\sigma(M) \mid L(M) \text{ обладает свойством } P\}.$$

Пример 2.

1. Если P — это “непустота”, то
$$L_P := \{\sigma(M) \mid L(M) \neq \varnothing\}.$$
Такой язык рекурсивно-перечислим.
2. $$L_P := \{\sigma(M) \mid M \text{ отвергает конечное число программ на C++}\}.$$
3. $$L_P := \{\sigma(M) \mid L(M) = L_0\}.$$
Как мы уже знаем $L_P = \varnothing$.

Теорема 5 (Райса)**.** *Всякое нетривиальное свойство неразрешимо.*