3. Семинар 21.02.2017

3.1. Решение задач из мини-контрольной

Задача 1. Пусть $f(n) = O\big(g(n)\big)$. Верно ли, что $2^{f(n)} = O\big(2^{g(n)}\big)$? Если ответ да, то надо доказать утверждение, если нет — привести контрпример.

Решение. **Нет, неверно.** Рассмотрим $f(n) = 2n$, $g(n) = n$. Очевидно, $f(n) = O\big(g(n)\big)$. Но при этом

$$\frac{2^{f(n)}}{2^{g(n)}} = 2^n \to \infty \text{ при } n \to \infty,$$

поэтому $2^{f(n)} \neq O\big(2^{g(n)}\big)$. □

Задача 2. Пусть про некоторую задачу известно, что любой алгоритм (МТ), решающий ее, имеет сложность $\Omega(n^2)$. Значит ли это следующее: существует такая константа c, что любой алгоритм A, решающий эту задачу, на любом входе x, длина которого больше некоторого числа n_A, будет делать не меньше $c|x|^2$ шагов? Если нет, то исправьте утверждение.

Решение. **Неверно.** Это означает, что для любого алгоритма A, решающего эту задачу, существуют такие $c_A > 0$, $n_A \in \mathbb{N}$, что для любого $n \geq n_A$ выполняется $T_A(n) \geq cn^2$, т.е. сущствует вход x длины n, на котором алгоритм A делает не меньше cn^2 шагов. □

Задача 3. Постройте МТ с тремя лентами, которая выполняет сложение чисел: на первую ленту подается число x в бинарной записи, на вторую — y, на третьей после окончания работы должно быть записано число $x + y$.

Решение. МТ будет иметь три состояния: q_0 означает, что в следующий разряд переносится 0 (также это стартовое состояние), q_1 — что переносится 1, $stop$ — финальное состояние. Функция переходов:

$$\begin{aligned}
&\delta(q_i, j, k, \sqcup) = (q_a, j, k, b, +1, +1, +1), \text{ где } a := \Big\lfloor \frac{i+j+k}{2} \Big\rfloor, \ b := (i+j+k) \bmod 2,\\
&\delta(q_i, j, \sqcup, \sqcup) = (q_a, j, \sqcup, b, +1, +1, +1), \text{ где } a := \Big\lfloor \frac{i+j}{2} \Big\rfloor, \ b := (i+k) \bmod 2,\\
&\delta(q_i, \sqcup, k, \sqcup) = (q_a, \sqcup, k, b, +1, +1, +1), \text{ где } a := \Big\lfloor \frac{i+k}{2} \Big\rfloor, \ b := (i+k) \bmod 2,\\
&\delta(q_0, \sqcup, \sqcup, \sqcup) = (stop, \sqcup, \sqcup, \sqcup, 0, 0, 0),\\
&\delta(q_1, \sqcup, \sqcup, \sqcup) = (stop, \sqcup, \sqcup, 1, 0, 0, 0)
\end{aligned}$$

(через $\lfloor x \rfloor$ будем обозначать целую часть числа x, т.е. округление вниз). □

3.2. Вычислимые функции

Во-первых, приведем пример разрешимого множества, для которого мы не укажем явного алгоритма.

Пример 1. Рассмотрим множество $A \subset \mathbb{N}$, состоящее из таких n, что в десятичной записи числа π есть n девяток подряд. На первый взгляд не понятно, является ли это множество разрешимым. Тем не менее, это так: действительно, либо $A = \mathbb{N}$, либо $A = \{0, 1, \ldots, n\}$ для какого-то n. В обоих случаях это множество разрешимо, хотя мы и не указали явно алгоритм, который бы его разрешал.

Из существования универсальной МТ следует, что существует **универсальная вычислимая функция**, т.е. такая функция $U(n, x)$, что для любой вычислимой функции $f(\cdot)$ найдется номер n, при котором $f(x) = U(n, x)$ для всех x, и если $f(x)$ не определена, то $U(n, x)$ тоже не определена. В то же время, *не существует универсальной всюду определенной вычислимой функции*. Действительно, допустим, что $V(n, x)$ — такая функция. Тогда функция $f(x) := V(x, x) + 1$ является вычислимой и всюду определенной, но она отличается от любого сечения $V(n, \cdot)$, т.е. и от любой всюду определенной вычислимой функции — противоречие. По этой же причине *не существует универсального разрешимого множества*, хотя существует универсальное перечислимое.

Напомню, что *busy beaver* функция $R(n)$ равна максимальному числу единиц, которое может быть записано на ленте после окончания работы МТ с n состояниями, запущенной на пустом входе. Как мы видели, эта функция невычислима. Более того, можно показать, что она *растет быстрее любой вычислимой функции*. Рассмотрим произвольную всюду определенную вычислимую функцию $f(\cdot)$. Она реализуется некоторой МТ M в унарном алфавите. Рассмотрим МТ M' с $3n+C$ состояниями, которая сначала записывает на ленту $4n$ единиц, а потом запускает на этом входе МТ M, и в итоге печатает $f(4n)$ единиц. Тогда в силу строгой монотонности функции $R(\cdot)$ при $n > C$ получаем

$$R(4n) > R(3n + C) \geq f(4n).$$

Следовательно, $R(n)$ растет быстрее $f(n)$.

3.3. Сводимость

Язык L_1 m**-сводится** к языку L_2 ($L_1 \leq_m L_2$), если существует такая всюду определенная вычислимая функция $f(\cdot)$, что

$$x \in L_1 \Leftrightarrow f(x) \in L_2.$$

Такая функция называется m-сводящей L_1 к L_2.

Свойства m-сводимости.

- Если L_2 — разрешимый, и $L_1 \leq_m L_2$, то L_1 тоже разрешимый. Действительно, характеристическая функция $\chi_{L_2}(\cdot)$ вычислима, поэтому $\chi_{L_1}(x) = \chi_{L_2}\big(f(x)\big)$ тоже вычислима.
- Если L_2 — перечислимый, и $L_1 \leq_m L_2$, то L_1 тоже перечислимый. Аналогично предыдущему случаю, полухарактеристическая функция $\tilde{\chi}_{L_1}(x) = \tilde{\chi}_{L_2}\big(f(x)\big)$ вычислима.
- Рефлексивность: $L \leq_m L$. Очевидно, в этом случае можно взять $f(x) \equiv x$.
- Транзитивность: если $L_1 \leq_m L_2$ и $L_2 \leq_m L_3$, то $L_1 \leq_m L_3$. Действительно, если $f(\cdot)$ m-сводит L_1 к L_2, а $g(\cdot)$ m-сводит L_2 к L_3, то их всюду определенная вычислимая композиция $g(f(\cdot))$ m-сводит L_1 к L_3.
- Если $L_1 \leq_m L_2$, то $\overline{L_1} \leq_m \overline{L_2}$. Очевидно, можно использовать ту же функцию $f(\cdot)$, которая m-сводит L_1 к L_2.

Перечислимый язык L называется m**-полным** [в классе перечислимых языков], если любой перечислимый язык L' m-сводится к L. В силу транзитивности, если m-полный язык L_1 m-сводится к перечислимому языку L_2, то L_2 тоже является m-полным. Заметим, что m-полный язык обязательно является *неразрешимым*, иначе все перечислимые языки были бы разрешимы, что неверно.

Приведем два примера m-полных языков.

Пример 2. Пусть $W \subset \mathbb{N} \times \Sigma^*$ — универсальное перечислимое множество. Очевидно, существует вычислимое кодирование пар, т.е. вычислимое взаимно-однозначное соответствие $\mathbb{N} \times \Sigma^* \ni (n, x) \leftrightarrow [n, x] \in \Sigma^*$ (аналогично, существует и вычислимая нумерация пар). Обозначим перечислимый язык, получаемый при данном кодировании множества W, как $\tilde{W}$. Покажем, что он m-полный. Рассмотрим произвольный перечислимый язык L. Т.к. W — универсальное множество, то существует такой номер n, что $L = W_n := \{x : (n, x) \in W\} = \{x : [n, x] \in \tilde{W}\}$. Следовательно, вычислимое отображение $f(x) := [n, x]$ m-сводит L к $\tilde{W}$, поэтому $\tilde{W}$ — m-полный язык.

Пример 3. Другим примером m-полного языка является язык L_{stop}, состоящий из описаний всех МТ, останавливающихся на пустом входе ε. Пусть L — некоторый перечислимый язык. Тогда существует принимающая его МТ M. Без ограничения общности будем считать, что M останавливается на входе x, только если $x \in L$ (состояние $Reject$ можно заменить на бесконечный цикл). Тогда для каждого слова x построим МТ M_x, которая, получив пустой вход, пишет на ленте слово x и запускает на нем машину M. Таким образом мы построили вычислимое отображение $f(x) := \langle M_x \rangle$, для которого, очевидно, выполняется

$$x \in L \Leftrightarrow \text{"}M \text{ останавливается на } x\text{``} \Leftrightarrow \text{"}M_x \text{ останавливается на } \varepsilon\text{``} \Leftrightarrow f(x) = \langle M_x \rangle \in L_{stop}.$$

Таким образом, $L \leq_m L_{stop}$.

Оказывается, что все m-полные языки *изоморфны*, а именно: для любых m-полных языков L_1, L_2 существует вычислимая биекция, переводящая их друг в друга, т.е. такая взаимно-однозначная вычислимая $\phi \colon \Sigma^* \to \Sigma^*$, что $x \in L_1 \Leftrightarrow \phi(x) \in L_2$.

Пусть P — некоторое свойство языка, т.е. предикат (высказывание) об языке, принимающий значения $True$ или $False$. Следующая теорема описывает некоторый класс неразрешимых задач, связанный со свойствами языков.

***Теорема* 1** (Успенский, Райс)**.** *Пусть свойство P не тривиальное, т.е. существуют такие перечислимые языки L_1, L_2, что $P(L_1)$, но $\neg P(L_2)$. Тогда задача определения по описанию МТ $\langle M \rangle$, верно ли $P\big(L(M)\big)$ (т.е. обладает ли язык, принимаемый M, свойством P) является алгоритмически неразрешимой.*

Доказательство. Без ограничения общности будем считать, что пустой язык $\emptyset$ не обладает свойством P, а язык $L(T)$ для некоторой МТ T — обладает. Пусть L_P состоит из описаний тех МТ, принимаемый которыми язык обладает свойством P. Покажем, что $L_{stop} \leq_m L_P$, следовательно, L_P — неразрешимый. Для произвольной МТ M построим МТ M', действующую следующим образом: получив на вход x, она сохраняет его на одной ленте, потом запускает машину M на пустой ленте. После остановки M (если, конечно, она останавливается) запускаем машину T на входе x. Таким образом, если M останавливается на пустом входе, то $L(M') = L(T)$, иначе $L(M') = \emptyset$. Следовательно, вычислимое отображение $\langle M \rangle \mapsto \langle M' \rangle$ m-сводит L_{stop} к L_P. □

Пример 4. Например, неразрешима следующая задача, выглядящая довольно тривиально: по описанию МТ $\langle M \rangle$ определить, не пуст ли язык $L(M)$, т.е. принимает ли она хоть одно слово.

m-сводимость в какой-то степени означает, что мы используем информацию о языке L_2, чтобы выполнять проверку принадлежности языку L_1. В то же время, заметим, что, вообще говоря, $\overline{L}$ может не сводится к L, т.е. мы используем информацию весьма ограниченным способом. Обобщением, которое позволяет обойти это ограничение, является понятие Тьюринговой сводимости (T-сводимости). Рассмотрим алгоритм, которая имеет доступ к *оракулу*, дающему ответы на вопросы о языке L, т.е. может обращаться к функции $\chi_L(\cdot)$ в ходе вычислений. Будем говорить, что язык L_1 **сводится по Тьюрингу** к языку L_2 ($L_1 \leq_T L_2$), если существует МТ, имеющая доступ к оракулу для L_2, которая разрешает язык L_1 (говорят, что она L_2-разрешает язык L_1). Заметим, что сама функция $\chi_{L_2}(\cdot)$ не обязательно является разрешимой. Если бы это было так, то обращение к этой функции можно было бы заменить на вызов подпрограммы, ее вычисляющей, и особого смысла вводить понятие T-сводимости не было бы. Поэтому нетривиальны те случаи, когда язык L_2 является неразрешимым.

Свойства T-сводимости.

- Если $L_1 \leq_m L_2$, то $L_1 \leq_T L_2$. Действительно, $\chi_{L_1}(x) = \chi_{L_2}\big(f(x)\big)$ для некоторой вычислимой $f(\cdot)$.
- $L \leq_T \overline{L}$. Очевидно, $\chi_L(x) = 1 - \chi_{\overline{L}}(x)$.
- Если L_2 — разрешимый, и $L_1 \leq_T L_2$, то L_1 тоже разрешимый. В этом случае обращение к функции $\chi_{L_2}(\cdot)$ можно заменить на подпрограмму.
- Транзитивность: если $L_1 \leq_T L_2$ и $L_2 \leq_T L_3$, то $L_1 \leq_T L_3$. Действительно, мы можем вычислить $\chi_{L_1}(x)$, обращаясь к функции $\chi_{L_2}(\cdot)$, для вычисления которой достаточно иметь возможность обращаться к функции $\chi_{L_3}(\cdot)$.

Заметим, что неперечислимый язык может T-сводиться к перечислимому, в отличие от m-сводимости. Действительно, пусть L — перечислимый неразрешимый язык. Тогда $\overline{L}$ — неперечислимый, и $\overline{L} \not\leq_m L$, но при этом $\overline{L} \leq_T L$.

3.4. Арифметическая иерархия

Множества (языки) можно отождествлять с предикатами (свойствами), обозначающими принадлежности множеству. Так, множество A эквивалентно свойству $A(x) =$ "x принадлежит A". В терминах свойств m-сводимость A к B означает, что для некоторой всюду определенной вычислимой функция $f(\cdot)$ выполняется

$$A(x) \Leftrightarrow B\big(f(x)\big).$$

Класс Σ_n состоит из свойств, которые можно представить в следующем виде:

$$A(x) \Leftrightarrow \underbrace{\exists y_1 \forall y_2 \exists y_3 \ldots}_{n\text{ раз}} R(x, y_1, \ldots, y_n),$$

где R — разрешимое свойство. С точки зрения множеств, A получается из разрешимого множества R последовательностью операций проекции и дополнения. Аналогично, класс Π_n состоит из свойств, которые можно представить в следующем виде:

$$B(x) \Leftrightarrow \underbrace{\forall y_1 \exists y_2 \forall y_3 \ldots}_{n\text{ раз}} R(x, y_1, \ldots, y_n).$$

Т.к. отрицания разрешимого свойства тоже разрешимо, то Σ_n состоит из свойств, которые являются отрицаниями свойств из Π_n, и наоборот:

$$\neg A(x) \Leftrightarrow \neg\exists y_1\forall y_2 \ldots R(x, y_1, \ldots, y_n) \Leftrightarrow \forall y_1\neg\forall y_2 \ldots R(x, y_1, \ldots, y_n) \Leftrightarrow \ldots$$
$$\ldots \Leftrightarrow \forall y_2\exists y_2 \ldots \neg R(x, y_1, \ldots, y_n) \in \Pi_n.$$

В частности, будем считать, что $\Sigma_0 = \Pi_0$ — класс всех разрешимых множеств.

Множества, состоящие из пар, взаимно-однозначно соответствуют обычным множествам, поэтому можно дать эквивалентное определение Σ_n и Π_n по индукции: $A \in \Sigma_n$, если

$$A(x) \Leftrightarrow \exists y R(x, y), \text{ где } R \in \Pi_{n-1},$$

и $B \in \Pi_n$, если

$$B(x) \Leftrightarrow \forall y R(x, y), \text{ где } R \in \Sigma_{n-1}.$$

В частности, Σ_1 — перечислимые свойства, а Π_1 — коперечислимые.

Свойства классов Σ_n и Π_n.

- Если $A, B \in \Sigma_n$, то $A \cup B \in \Sigma_n$, $A \cap B \in \Sigma_n$; то же верно и для Π_n: если $A, B \in \Pi_n$, то $A \cup B \in \Pi_n$, $A \cap B \in \Pi_n$.

- Если $A \leq_m B$ и $B \in \Sigma_n[\Pi_n]$, то $A \in \Sigma_n[\Pi_n]$. Действительно,

  $$A(x) \Leftrightarrow B(f(x)) \Leftrightarrow \exists y_1\forall y_2 \ldots R(f(x), y_1, \ldots, y_n) \Leftrightarrow \exists y_1\forall y_2 \ldots \tilde{R}(x, y_1, \ldots, y_n),$$

  где $\tilde{R}(x, y_1, \ldots, y_n) \Leftrightarrow R(f(x), y_1, \ldots, y_n)$ — разрешимое свойство, т.к. R разрешимо, а функция $f(\cdot)$ вычислима.

- В каждом классе Σ_n и Π_n, для $n \geq 1$, есть универсальное для этого класса множество (аналогично перечислимым множествам). Покажем это на примере Π_2. Рассмотрим универсальное перечислимое множество троек W. Свойство $B \in \Pi_2$ имеет вид $B(x) \Leftrightarrow \forall y V(x, y)$, где V — перечислимое множество пар, которое для некоторого n равно $W_n := \{(x, y) : (n, x, y) \in W\}$. Следовательно,

  $$B(x) \Leftrightarrow \forall y W(n, x, y) \Leftrightarrow T(n, x),$$

  где $T(n, x) \Leftrightarrow \forall y W(n, x, y)$. Таким образом, T — универсальное множество в классе Π_2. Очевидно, его дополнение будет универсальным в Σ_2. Для всех остальных n универсальные множества строятся аналогично.

- В каждом классе Σ_n и Π_n есть m-полное множество. Для Σ_0, очевидно, любой нетривиальный разрешимый язык будет m-полным. В остальных классах m-полными являются универсальные множества.

- Для всех $n \geq 1$ выполняется $\Sigma_n \cup \Pi_n \subsetneq \Sigma_{n+1} \cap \Pi_{n+1}$. Отсюда также следует, что никакое m-полное в Σ_n множество не принадлежит Π_n, и наоборот.

Классы $\Sigma_0 = \Pi_0$, Σ_1, Π_1, Σ_2, Π_2 и т.д. образуют **арифметическую иерархию**. Чем больше n, тем более сложно устроенные множества могут содержать эти классы.

Заметим, что если некоторое множество является m-полным в Σ_n или Π_n для $n \geq 2$, то оно точно не является ни перечислимым, ни коперечислимым. Это дает способ доказывать неперечислимость множества, что мы пока что умели делать только через теорему Поста (т.е. для коперечислимых неразрешимых множеств) или с помощью явного построения.

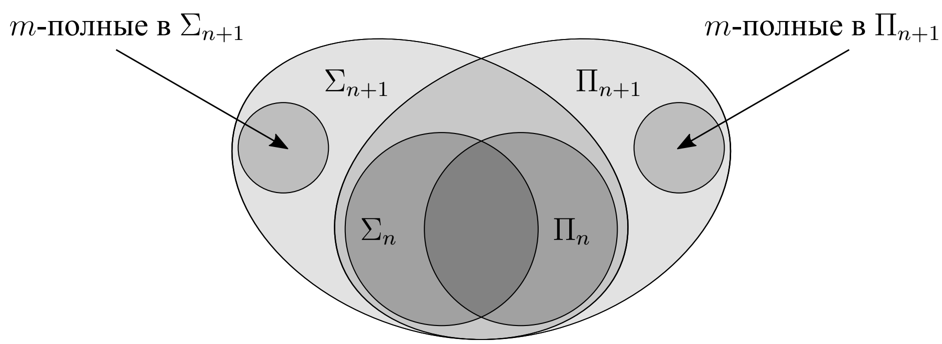

Рис. 1: Арифметическая иерархия