1 Билет 1

Утверждение 1.1. *Пусть M – одноленточная МТ, которая распознает язык бинарных палиндромов. Тогда существует константа $C : \exists n_o : \forall n > n_0$ существует вход длины n, на котором $M(x)$ делает $\geq Cn^2$ шагов.*

Доказательство. В начале очевиден принцип несжимаемости, нельзя инъективно перевести строки из $\{0,1\}^n$ в $\{0,1\}^*$ так, чтобы все образы по длине были меньше чем n.
Будем доказывать для входов, длина которых кратна 3. По $x \in \{0,1\}^n$ строим вход $x0^n x^{rev}$ и скармливаем МТ все такие входы. Возьмем все перегородки после нулей, их всего n, существует перегородку, через которую МТ прошла $\leq \frac{T(x)}{n}$ раз.
Теперь строим отображение $f : \{0,1\}^n \to \{0,1\}^*$, переводим строку x в протокол работы МТ на строке $x0^n rev(x)$. Для этого выпишем набор состояний, в которые переодила МТ, переходя через "хорошую"перегородку и номер этой перегородки.
Утверждается, что такая f – инъекция, чтобы доказать, предположите обратное и рассмотрите работу на строке $x0^n y^{rev}$.
Пусть $|x| = n$, тогда $f(x) \leq \log n + \frac{T(x)}{n}C$, но при этом существует $|y| = n$, такой что $f(y) \geq n$. Получаем, что

$$n \leq logn + \frac{T(x)}{n}C$$

$T(x) = \omega(n^2)$ на таких входах.
Для некратных 3 входов делаем также, но по-середине пишем вместо n нулей, на один ноль больше или меньше – это не влияет на оценки. ■

2 Билет 2

Определение 2.1. *k – ленточная машина Тьюринга. (Добавляется куча лент и функция перехода теперь действует по всем лентам).*

Утверждение 2.1. *Для любой k – ленточной МТ, которая на входе x работает время $T(x)$, существует 1 ленточная МТ, которая работает $O(T(x)^2)$.*

Доказательство. Будем хранить в одном символе МТ символы всех лент (а также спец символы, помеченные головкой). На каждом шаге будем идти вправо и делать все изменения, которые нужны на лентах. ■

Определение 2.2. *Универсальная МТ – эмулирует МТ по описанию.*

Утверждение 2.2. *Для любой k-ленточной МТ существует универсальная k-ленточная МТ с линейным замедлением.*

Доказательство. Понятно как получить квадратичное замедление, нужно положить описание в начало, например, первой ленты. Далее постоянно возвращаться, чтобы узнать, какой шаг сделать. Если же хотим линейного – давайте возить описание с собой, это будет давать $O(1)$ действий из-за его константного размера, при этом эмуляция будет работать за линейное время. ■

Утверждение 2.3. *k ленточную МТ можно эмулировать на 2-ленточной с логарифмическим замедлением.*

Доказательство. TODO ■

3 Билет 3

Основная модель вычислений – многоленточная МТ.

Определение 3.1. *$f : \mathbb{N} \to R_+$, тогда $L \in DTime[f(n)]$, если существует многоленточная МТ, такая что*

1. $\forall x \in L \Rightarrow M(x) = 1$.
2. $\forall x \notin L \Rightarrow M(x) = 0$.
3. $\forall x$ *МТ работает* $O(f(|x|)$ *шагов.*

Определение 3.2. $P = \cup_{i>0} DTime[n^i]$.

Определение 3.3. *Про семейство схем, распознающих язык.*

Определение 3.4. *$L \in Size[f(n)]$, если есть последовательность схем, распознающих L и для достаточно больших n выполнено $|C_n| \leq f(n)$.*

Определение 3.5. $P/Poly = \cup_{i>0} Size[n^i]$.

Пример 3.1. *Неразрешимый язык может лежат в $P/Poly$. Например $1^H = \{1^n | n \in H\}$, для некоторого языка тоже является разрешимым и лежит в $P/Poly$, так как на каждую длину мы можем предоставить схему.*

Утверждение 3.1. *Существует такой алгоритм A, который получает на вход T, n, m и*

1. *A работает $poly(n + T + |m|)$ шагов.*
2. *Если МТ m на всех входах из $\{0,1\}^*$ выдает ответ за $\leq T$ шагов, то алгоритм A выдает схему C, которая имеет n входов и 1 выход и распознает на входах длины n также как m.*

Доказательство. Будем возвращать схему размера $T \times T \cdot O(1)$.
На уровне i будет T ячеек, в каждой из которых будет вычисляться некоторая информация: сивмол, написанный в этой ячейке, есть ли тут головка в момент i, а также, если есть головка, то состояние, в которой МТ сейчас находится. Понятно, что для пересчета этих параметров нужно обратиться к нескольким соседним ячейкам предыдущей строки. Для того, чтобы узнать ответ, посмотрим, принималось ли где-нибудь состояние q_{yes}.

■

Утверждение 3.2. $P \subseteq P/Poly$.

Таким образом хотели доказывать, что $P \neq NP$, взять, к примеру, SAT и показать, что он не лежит в $P/Poly$, однако доказывать нижние оценки на схемы пока что не научились.