Листок 11. Теория сложности, продолжение.

Определение 1 **EXP** — *класс языков, разрешимых на ДМТ за время* $2^{\text{poly}(n)}$. **NEXP** — *класс языков, разрешимых на НМТ (для которых существует алгоритм проверки сертификата) за время* $2^{\text{poly}(n)}$.

Определение 2 *Пусть* A — *класс языков. Класс* $\mathbf{P}^A$ — *класс языков, для которых существует полиномиальный детерминированный алгоритм, который может обращаться к оракулу из класса* A.

ML 59. Покажите, что если $\mathbf{P} = \mathbf{NP}$, то $\mathbf{EXP} = \mathbf{NEXP}$.

ML 60. Докажите, что существует язык, для которого любой алгоритм, работающий время $O(n^2)$ решает его правильно на менее, чем на половине входов какой-то длины, но этот язык распознается алгоритмом, работающим время $O(n^3)$.

ML 61. Докажите, что:

(а) $\mathbf{DSpace}[n^3] \not\subseteq \mathbf{DSpace}[n^2]$;

(б) $\mathbf{NSpace}[n^3] \not\subseteq \mathbf{NSpace}[n^2]$.

ML 62. Унарным назвается язык, все слова которого состоят из одного символа. Докажите, что если все унарные языки из **NP** лежат в **P**, то $\mathbf{EXP} = \mathbf{NEXP}$.

ML 63. Пусть существует **NP**-полный унарный язык (все слова которого, состоят только из одного символа). Докажите, что $\mathbf{P} = \mathbf{NP}$ (подсказка: придумайте алгоритм для задачи SAT).

ML 64. Покажите, что:

(а) $\mathbf{P}^{\mathbf{P}} = \mathbf{P}$;

(б) язык GNI (пар неизоморфных графов) лежит в $\mathbf{P}^{\mathbf{NP}}$.

ML 65. Покажите, что:

(а) $\mathbf{P} \subseteq \mathbf{NP} \cap \mathbf{coNP}$;

(б) $\mathbf{NP} \subseteq \mathbf{EXP}$.

ML 23.

Задача Поста состоит в следующем: есть доминошки n видов $\left[\frac{s_1}{t_1}\right]$, $\left[\frac{s_n}{t_n}\right]$, s_i и t_i — конечные строки, есть неограниченный запас доминошек каждого вида, доминошки переворачивать нельзя. Требуется определить, можно ли составить несколько доминошек так, чтобы в верхней и нижней их половине читалась одна и та же строка, такие последовательности доминошек будем называть согласованными. Докажите, что задача Поста алгоритмически неразрешима.

$\boxed{\textbf{ML 33.}}$ Теперь секвенцией будем называть $\Gamma \vdash \Delta$, где Γ и Δ — это списки предикатных формул.

Добавим в секвенциальное счисление четыре новых правила которые соответствуют кванторам (см. табличку).

В правилах $(\forall\vdash)$ и $(\vdash\exists)$, $A(t/x)$ обозначает, что в формуле A переменная x заменяется на терм t, при этом для каждого вхождения переменной x никакие переменные терма t не должны попасть в область действия кванторов по одноименным переменным (в формуле A). Например для формулы $\forall y\ P(x, y)$ вместо x нельзя подставить $f(y)$.

А в других двух правилах $A(y/x)$ означает, что в формуле A мы заменили все вхождения x на переменную y, при этом переменная y должна быть свежей то есть не входить ни в A, ни в другие формулы из секвенции.

Докажите корректность секвенциального исчисления (покажите, что если секвенция $\Gamma \vdash \Delta$ выводима, то в любой интерпретации либо хотя бы одна формула из Γ ложна, либо хотя бы одна формула из Δ истинна).

$\boxed{\textbf{ML 40.}}$ Пусть T — замкнутая формула в некоторой сигнатуре, и пусть существует интерпретация со сколь угодно большим носителем, в которой данная формула истинна. Докажите, что существует интерпретация с бесконечным носителем, в которой данная формула истинна.

$\boxed{\textbf{ML 50.}}$ Будет ли теория $\mathrm{Th}((\mathbb{N}, <, =))$ конечно аксиоматизируемой.

$\boxed{\textbf{ML 55.}}$ Пусть функции $f, g : \{0,1\}^* \to \{0,1\}^*$ можно посчитать с использованием $O(\log(n))$ памяти (память считается только на рабочих лентах, входная лента доступна только для чтения, а по выходной ленте головка машины Тьюринга движется только слева направо). Докажите, что функцию $f(g(x))$ можно также посчитать с использованием $O(\log(n))$ памяти.

$\boxed{\textbf{ML 58.}}$ Докажите, что:

(a) что число n простое тогда и только тогда, когда для каждого простого делителя q числа $n-1$ существует $a \in \{2, 3, \ldots, n-1\}$ при котором $a^{n-1} \equiv 1 \pmod{n}$, а $a^{\frac{n-1}{q}} \not\equiv 1 \pmod{n}$;