Домашнее задание 4
Турков Матвей, группа 777

(1)

Решение:

1. Так как разница между ДМТ и НДМТ лишь в функции переходов δ, то мы можем повторить алгоритм доказательства замнкнутости P, но уже на НДМТ, что, по определению, даст замкнутость NP-языков по итерации.

2. Для любого языка $L \in P$ верно, что $\bar{L} \in P \Rightarrow \bar{L} \in NP \Rightarrow L \in co - NP$.

(2)

Решение:

1. Нашим сертификатом будет набор значений, на которых формула выполнима. Верификатор - функция, которая будет выполнять следующий алгоритм: идем до первой закрывающей скобки, после нее возвращаемся назад, подставляя в пропозицонные символы значения и, встретив открывающую скобку, вычисляем значение скобки и заменяем ее на нужную константу. Каждый раз, сворачивая скобку, мы уменьшаем формулу минимум на 1 символ/связку. Всего проходов может быть $O(n^2)$. Следовательно, данная функция полиномиальна.

2. В виде сертификата предоставим вектор вершин размера $|V|$(1 - вершина входит, 0 - не входит). Далее проходом по графу, просматривая все ребра, проверяем данный сертификат.

3. Сертификат - посл-ть вершин эйлерова пути в данном графе. Проходим один раз по графу, проверяя условие эйлерова пути и получаем ответ.

4. Применяя алгоритм Форда-Беллмана мы за полином удостоверимся в наличии или отсутствии цикла с отрицательными весами ($\in P$). Если после $n - 1$ фазы мы выполним ещё одну фазу, и на ней произойдёт хотя бы одна релаксация, то граф содержит цикл отрицательного веса, достижимый из v; в противном случае, такого цикла нет.

5. Любую КС-грамматику можно за полином свести к НФХ(см. 2-ое ДЗ). Далее, используя алгоритм Кока-Янгера-Касами узнаем выводимость слова ($\in P$).

6. По теореме Понтрягина-Куратовского граф планарен т.к. он не содержит подграфов, гомеоморфных полному графу из пяти вершин K_5 или графу «домики и колодцы» $K_{3,3}$. Т.к. данная теорема - критерий, то $L_{planarity} \in NP \bigcap co - NP$. Следовательно, доказав принадлежность какому-то из классов, сразу докажем и принадлежность другому.

   Сертификат некомпланарности - набор вершин подграфа Куратовского. Его же вычисление может быть сделано методом добавления пути/вершин/ребер (см. Википедия). Все эти алгоритмы работают за полином, следовательно, $L_{planarity} \in P$.

3

Решение:

$$L \in co - NP \Rightarrow \bar{L} \in NP$$

1. $\bar{L}$ - нетавтологичные формулы. Следовательно, сертификат - вектор значений, на котором формула ложна. Далее, аналогично задаче $3 - SAT$, получаем ответ за полином.

2. $\bar{L}$ - описания графов, таких, что $\exists k : \forall (v_i, v_j) \in E$???Ненорм условие описано

   Сертификат - вектор ребер с k единицами. Проходя по графу один раз мы проверяем требуемое условие за полином.

3. Сертификат - массив простых делителей p_i числа a. Проверяем принадлежность хотя бы одного из них отрезку $[b, c]$.

4. Необходимым и достаточным условием для существования клики размера k является наличие независимого множества размера не менее k в дополнении графа. Сертификат - вектор вершин дополнения графа, входящих в независимое подмн-во размера не меньше k. Далее проверяем условие клики.

5. Смотри 2(6).

④

Решение:

$$L \in co - NP \Rightarrow \bar{L} \in NP$$

L - язык общезначимых предикатов, тогда $\bar{L}$ - необщезначимые предикаты $\rightarrow \exists x : g(x) = 0$.

Предоставляя в виде сертификата вектор значений пропозициональных символов мы, аналогично задаче $2(i)$ проходим по предикату, сводя его к константе. Следовательно, $L \in co - NP$.

⑤

Решение:

Суммарная длина простых делителей $p - 1$ не превышает $\log p$. Самих уровней в дереве - не более $\log p$: минимальный простой делитель 2. Для нахождения порождающего элемента быстро возводим в степень за $O(\log p)$ операций умножения и $O(\log p)$ операций взятия остатка, сложность каждой операции $O(\log^2 p)$, поэтому за $O(\log^3 p)$ можно подсчитать k^{p-1}, где k - порождающий элемент. Всего чисел $k^{\frac{p-1}{p_i}}$(, где p_i - простые делители числа $p-1$) $O(\log p)$, значит проверить истинность всех можно за $O(\log^4 p)$. Проверка того, что простые делители $p - 1$ есть p_i и только они делается за $O(\log^3 p)$. Поэтому:

$$T(p) = O(\log p) + \sum_{i=1}^{k} T(p_i)$$

Т.к. $x_1^k + \cdots + x_n^k \leq (x_1 + \cdots + x_n)^k$, то сумму на каждом уровне оценим в $O(\log^4 p)$, всего уровней $\log p$. Получаем, что сложность проверки сертификата $O(\log^5 p) \Rightarrow$ полиномиальна.

$$100091237, 100091236 = 2^2 * 7 * 3574687, k = 2$$

$$7, 6 = 2 * 3, k = 2$$

$$3574687, 3574686 = 2 * 3 * 233 * 2557, k = 62$$

$$233, 232 = 2^3 * 29, k = 2$$

$$29, 28 = 2^2 * 7, k = 2$$

$$2557, 2556 = 2^2 * 3^2 * 71, k = 2$$

$$71, 70 = 2 * 5 * 7, k = 14$$

6

Решение:

Ссылки на решение задач One, Two

7

Решение:

Язык L лежит в классе NP, следовательно, для него существует функция $V(x, s)$ с булевыми значениями, вычислимая за полиномиальное время от длины первого аргумента, такая, что:

$$x \in L \to \exists s : V(x, s) = 1;$$
$$x \notin L \to \forall s : V(x, s) = 0;$$

1. На вход алгоритму приходит описания 2-ух графов. Требуется проверить их на изоморфность. Следовательно, нужно предоставить сертификат полиномиального размера от длины входа.

   Сертификат - две строки массивов длины $|V|$, верхняя строка - вершины в произвольном (заданном) порядке, вторая - куда они отображаются. Далее, пройдя по графу, проверяем наличие путей в изначальном графе и в изменном по функции-сертификату.

2. Сертификат - размер клики k. Далее проходом по графу мы проверяем каждую вершину, есть ли от нее k ребер. Записываем им в определенный массив и на следующем шаге (переходом по одному из ребер) проверяем, будет ли от следующей вершины тоже k ребер. Если из вершины нет k ребер, то удаляем ее из списка вершин. Таким образом мы сведем граф к полному.

3. Приведем систему к ступенчатому виду. Далее, если система имеет экзотические уравнения ($Click$). Это мы сделаем за полином и по критерию найдем ответ.