Домашнее задание по АМВ

771 группа, Христолюбов Максим

9 марта 2019 г.

1 Задача 0

$x = 13, 13^3 910 \equiv 1 \mod 3911$

$3910 = 2 \cdot 5 \cdot 17 \cdot 23$, $x^{\frac{3910}{p_i}} \not\equiv 1 \mod 3911$

Найдем x, такой что $x^{16} \equiv 1 \mod 17$ и $x^{\frac{16}{p_i}} \not\equiv 1 \mod 17$:

$6^{16} \equiv 36^8 \equiv 2^8 \equiv 16^2 \equiv (-1)^2 \equiv 1 \mod 17$, как видно $6^{\frac{16}{p_i}} \not\equiv 1 \mod 17$.

У 16 один делитель 2 и он простой.

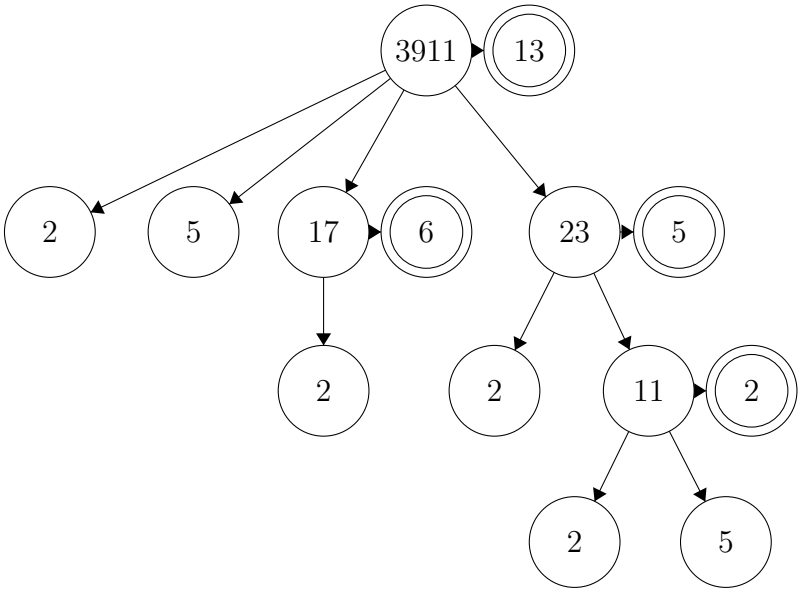
Найдем x, такой что $x^{22} \equiv 1 \mod 23$ и $x^{\frac{22}{p_i}} \not\equiv 1 \mod 23$:

$5^{22} \equiv 25^{11} \equiv (-2)^{11} \equiv 1 \mod 23$, $5^{11} \equiv 22 \mod 23$.

У 22 два делителя 2 и 11. Проверим простоту 11. Найдем x, такой что $x^{10} \equiv 1 \mod 11$ и $x^{\frac{10}{p_i}} \not\equiv 1 \mod 11$:

$2^{10} \equiv 32^2 \equiv (-1)^2 \equiv 1 \mod 11$

В итоге получится дерево, которое является сертификатом. Дочерние вершины с числом n это делители $n-1$, а приставленная к n вершина в двойном круге это число порядка $n-1$ в кольце вычетов по модулю n.



2 Задача 1

2.1 (i)

$\varphi \in L \Leftrightarrow \exists\overline{x}\forall\overline{y} \mapsto \varphi(\overline{x},\overline{y}) = 1$, и по определению Σ_2 $L \in \Sigma_2$.

2.2 (ii)

В Σ_3 лежит задача распознавания языка L из булевых формул $\varphi = \varphi(\overline{x},\overline{y},\overline{z})$, таких что существует набор $\overline{x}$ такой, что для всех $\overline{y}$ существует $\overline{z}$, такой что $\varphi(\overline{x},\overline{y},\overline{z}) = 1$.

2.3 (iii)

Проверять за полиномиальное время, что $\exists y_1\forall y_2 \dots K_k y_k \mapsto R(x,y_1 \dots y_k)$ (K_k — последний квантор), можно если есть возможность проверить полиномиально $\exists y_1\forall y_2 \dots K_k y_k \overline{K_k} v \mapsto R'(x,y_1 \dots y_k,v)$, где v — просто фиктивная переменная, которая никак не влияет на результат, значит, $\Sigma_k \subset \Sigma_{k+1}$. С другой стороны это так же можно проверять, если есть возможность проверить $\forall v\exists y_1\forall y_2 \dots K_k y_k \mapsto R''(x,v,y_1 \dots y_k)$, значит, $\Sigma_k \subset \Pi_{k+1}$. Следовательно, $\Sigma_k \subset \Sigma_{k+1} \cap \Pi_{k+1}$

2.4 (iv)

Так как задачи из $\mathcal{NP}$ выполняются на недетерминированной МТ за полиномиальное время, то в ходе работы просто не может быть задействовано более $pol(n)$ ячеек, значит, $\mathcal{NP} \subset \mathcal{NPSPACE} = \mathcal{PSPACE}$ из Savitch's theorem.

В МТ конечное число q состояний, поэтому у МТ с полиномиальной лентой будет не больше $q \cdot 2^{pol(n)}$ конфигураций. МТ не может побывать в одной конфигурации дважды, так как переход МТ в конфигурацию, которая уже была, означает наличие цикла в работе МТ, что невозможно, поскольку МТ дает ответ. Значит всего переходов не более $q \cdot 2^{pol(n)}$ и затрачиваемое время экспоненциально, следовательно, $\mathcal{PSPACE} \subset \mathcal{EXPTIME}$.

Итак, $\mathcal{NP} \subset \mathcal{PSPACE} \subset \mathcal{EXPTIME}$.

3 Задача 2

Составим матрицу $n \times n$, такую что $a_{j_p i_p} = 0$, а все остальные элементы равны 1.

$Perm(A) = \sum\limits_{\pi \in S_n} a_{1\pi_1} a_{2\pi_2} \dots a_{n\pi_n}$

Тогда каждому слагаемому матрицы соответствует перестановка своя перестановка $\pi \in S_n$. Если в перестановке есть i_p, которая занимает место j_p, то в соответствующем слагаемом будет присутствовать $a_{j_p i_p} = 0$ и слагаемое будет равно 0. Если же никакие элементы не заняли запрещенные места, то все множители в слагаемом будут равны 1, и само слагаемое 1. Таким образом сумма, которой равен перманент, будет подсчитывать кол-во подходящих нам перестановок. Значит, можно найти кол-во таких перестановок решив задачу о нахождении перманента этой матрицы.

4 Задача 3

Длина построенной ДНФ может не зависеть полиномиально от длина КНФ. Например, для КНФ вида $(x_{11} \vee \dots \vee x_{1n}) \wedge \dots \wedge (x_{n1} \vee \dots \vee x_{nn})$, длина которой $\Theta(n^2)$, длина ДНФ будет не полиномиальна от n^2. Чтобы получить ДНФ нужно раскрыть все скобки этого выражения, причем после этого никакие конъюнкции ДНФ не могут быть выброшены, так как во всех них разные наборы переменных, и все они влияет на значение

формулы. После раскрытия скобок в ДНФ окажется n^n конъюнкций из n элементов, и длина ДНФ будет не полиномиальна от n^2.

5 Задача $3\frac{1}{2}$

Полнота задачи о выполнимости КНФ была рассмотрена на семинаре (через машину Тьюринга).

Задача о выполнимости ДНФ из $\mathcal{P}$, так как для того чтобы ДНФ была верна требуется чтобы все литералы из хотябы одного дизъюнкта были равны 1. То есть нужно для каждого конъюнкта, размер которого полиномиален от длины ДНФ проверить не присутствуют ли в нем литерал и отрицание этого литерала. Если присутствует, то этот дизъюнкт не выполним, переходим к следующему, если же отсутствует, то дизъюнкт выполним, а значит и вся ДНФ выполнима. Проверка занимает полиномиальное время.

Задача о тавтологичности КНФ из $\mathcal{P}$, так как для этого требуется проверить, что каждый из дизъюнктов на всех наборах равен 1, то есть все литералы в дизъюнкте не могут одновременно равняться 0. Если в дизъюнкте встречается литерал и его отрицание, то все литералы не могут равняться 0, если же не встречается, то существует набор, на котором все литералы из этого дизъюнкта равны 0, и КНФ не тавтологична. Проверка для каждого дизъюнкта есть ли в нем литерал и его отрицание можно за полиномиальное время.

Опровергнуть тавтологичность ДНФ можно предоставив в качестве сертификата набор, на котором ДНФ равна 0. Значит, задача об опровержении тавтологичности ДНФ в $\mathcal{NP}$, а задача об подтверждении тавтологичности ДНФ $co-\mathcal{NP}$. Кроме того $co-\mathcal{NP}-$с задачу об опровержении выполнимости КНФ можно свести по Карпу к задаче подтверждении тавтологичности ДНФ: в качестве $f(x)$ можно взять формулу, полученную из x заменой конъюнкций на дизъюнкции и наоборот. Действительно, если на всех наборах ДНФ $f(x)$ равна 1 (то есть на всех наборах существует дизъюнкт D_j, равный 1, что значит что в нем есть литерал $t_i = 1$), то на всех противоположных наборах КНФ x будет равен 0 (так как на всех противоположных наборах будет конъюнкт D_j, в котором есть литерал $t_i = 0$, будет равен 0). Следовательно, на всех наборах КНФ не равен 1 и не выполним. В другую сторону это так же выполнено. Значит, задача подтверждении тавтологичности ДНФ $co-\mathcal{NP}-$с, поскольку она сводится к задаче $co-\mathcal{NP}-$с.

	выполнимость	тавтологичность
КНФ	$\mathcal{NP}-c$	$\mathcal{P}$
ДНФ	$\mathcal{P}$	$co-\mathcal{NP}-c$

6 Задача 4

$\int\limits_1^{n+1}(x-1)^{\frac{1}{2}}dx \le \sum\limits_{k=1}^{n} k^{\frac{1}{2}} \le \int\limits_1^{n+1} x^{\frac{1}{2}}dx$

$\frac{2}{3}((n+1-1)^{\frac{3}{2}}-0) \le \sum\limits_{k=1}^{n} k^{\frac{1}{2}} \le \frac{2}{3}((n+1)^{\frac{3}{2}}-1)$

$\sum\limits_{k=1}^{n} k^{\frac{1}{2}} = \Theta(n^{\frac{3}{2}})$

$\int\limits_1^{n+1}(x-1)^{\alpha}dx \le \sum\limits_{k=1}^{n} k^{\alpha} \le \int\limits_1^{n+1} x^{\alpha}dx$

$\frac{1}{\alpha+1}((n+1-1)^{\alpha+1}-0) \le \sum\limits_{k=1}^{n} k^{\alpha} \le \frac{1}{\alpha+1}((n+1)^{\alpha+1}-1)$

$\sum\limits_{k=1}^{n} k^{\alpha} = \Theta(n^{\alpha+1})$

7 Задача 5

7.1 а)

Да, она останется полной, так как 3-SAT без дополнительных ограничений можно будет свести по Карпу к формулам из 3-SAT, в которых каждая переменная входит не более 3 раз, а каждый литерал — не более 2 раз.

Для этого в $\phi \in$ 3-SAT какая либо переменная x встречается n раз, то заменим ее на n новых переменных, и добавим к ϕ конъюнкцию $k_x = (x_1 \vee \overline{x}_2) \wedge (x_2 \vee \overline{x}_3) \wedge (x_3 \vee \overline{x}_4) \wedge \ldots \wedge (x_n \vee \overline{x}_1)$, которая является истинной тогда и только тогда, когда $x_1 = x_2 = \ldots = x_n$. Сделав это для каждой переменной получим $f(\phi) = \phi \wedge k_1 \wedge k_2 \wedge \ldots \wedge$
$\wedge\, k_m \in$ 3-SAT-ИЗ-УСЛОВИЯ, которая обеспечивает сводимость. Действительно, в части $f(\phi)$, которая была получена из ϕ избавлением от повторных переменных, каждая переменная встречается только 1 раз. Каждая переменная входит в добавленные конъюнкции k_i ровно 2 раза, а каждый литеарли 1 раз. Значит, $f(\phi)$ удовлетворяет условию. Сводимость построена.

8 Задача 6

По графу x, в котором ищется k-клика с n вершинами построим граф $f(x)$:

Если $n \leq 2k$, то $f(x) = x$, если $n > 2k$, то $f(x)$ получим добавив к x $(n - 2k)$ вершин, соединенные между собой и со всеми вершинам графа x. В первом случае k-клика уже на хотябы половине вершин и эквивалентность при $f(x) = x$ очевидна.

Рассмотрим второй случай, когда $n > 2k$, тогда в $f(x)$ $(2n - 2k)$ вершин. Каждой максимальной по включению p-клике из x будет соответствовать максимальная по включению $(p + n - 2k)$-клика из $f(x)$. Действительно, новых ребер между старыми вершинам не появилось, поэтому если к p-клике из x нельзя было добавить еще одну вершину, то к соответствующей $(p + n - 2k)$-клике из $f(x)$ тоже нельзя будет добавить вершину.

Значит, если в x есть клика размера k, то в $f(x)$ есть клика размером $k + n - 2k = n - k$, что равно половине от числа вершин $f(x)$. Если же в $f(x)$ есть клика на хотябы половине вершин, то есть $(p + n - k)$-клика, $p \geq 0$, то в x есть клика на $p + n - k - (n - 2k) = p + k$ вершинах. Взяв любые k вершин из этой клики получим k-клику. Значит, в этом случае в x есть k-клика. Итак эквивалентность $x \in A$ и $f(x) \in B$ доказана и сводимость построена.

*Идея добавить к x сколько-то вершин, соединенными со всеми остальными вершинам, принадлежит Юрию Прохорову.

9 Задача 7

9.1 а)

Да, например $f(n) = n^{\log_2 n}$, так как обязательно $\exists N : \forall n > N \mapsto \log_2 n > c \ \forall c > 0$, с другой стороны, $n^{\log_2 n} = 2^{\log_2^2 n}$ и $\exists N : \forall n > N \mapsto \log_2^2 n < dn \ \forall d > 0$. Значит, $n^{\log_2 n} = \omega(n^c) = o(2^{dn})$

9.2 б)

Да, тогда $P \neq co - \mathcal{NP}$.

Задача на проверку тавтологичности формулы формата 4-ДНФ лежит в $co - \mathcal{NP}$, так как предоставив в качестве сертификата набор, на котором формула не равна 1, подставив его в ДНФ можно полиномиально проверить, что формула не лежит в этом языке.

Но если любой МТ требуется $\Omega(n \log_2^{\log_2 n} n)$, тогда эта задача не лежит в $\mathcal{P}$, поскольку для любого c:

$\lim\limits_{n\to\infty}(\frac{n^c}{n \log_2^{\log_2 n} n}) = \lim\limits_{n\to\infty}(\frac{2^{c-1}}{\log_2 n})^{\log_2 n} = \lim\limits_{n\to\infty}(\frac{2^{c-1}}{2^{\log_2 \log_2 n}})^{\log_2 n} = \lim\limits_{n\to\infty} 2^{(c-1-\log_2 \log_2 n)\log_2 n} =$
$\lim\limits_{n\to\infty} n^{c-1-\log_2 \log_2 n} = \lim\limits_{n\to\infty}(\frac{1}{n})^{\log_2 \log_2 n - c + 1} = 0$

Значит, $n \log_2^{\log_2 n} n$ растет быстрее любого полинома, и всякий алгоритм, определяющий тавтологичность формул будет работать дольше, чем полиномиальное время. Получили, что эта задача лежит в $co - \mathcal{NP}$ и не лежит в $\mathcal{P}$, а значит это различные классы.