Рябых Владислав, Б05-905

Алгоритмы и модели вычислений. Домашнее задание № 6

Задача 1. Теорема Рейнгольда утверждает, что язык $UPATH = \{(G,s,t) |$ в неориентированном графе G есть неориентированный путь из s в $t\}$ лежит в $\mathcal{L}$. Опираясь на этот факт, докажите, что следующие языки лежат в $\mathcal{L}$:

1. $EVENCONN = \{G |$ неориентированный граф G имеет чётное число компонент связности$\}$;
2. $EDGEUCYCLE = \{(G,e) |$ в неориентированном графе G существует цикл, содержащий ребро $e\}$;
3. $XOR2SAT = \{\varphi \mid \varphi$ — конъюнкция выражений вида $x_i \oplus x_j$, для которой есть выполняющий набор$\}$.

Решение.

1. Для определения, лежит ли граф G в нашем языке $EVENCONN$, будем хранить в памяти всего 2 счётчика: счётчик текущей вершины num и бит, отвечающий за чётность числа компонент $IsEven$ (изначально равен 1). Мы нумеруем все вершины графа и действуем следующим образом, пока num не пройдёт все вершины нашего графа.

   - Если из 1 вершины есть путь в вершину num (что мы проверяем за логарифмическую память по теореме Рейнгольда), то мы увеличиваем счётчик num: $num = num + 1$.
   - Если же из 1 вершины нет пути в вершину num (что мы проверяем за логарифмическую память по теореме Рейнгольда), то мы проверяем наличие пути из num в какую-нибудь из вершин с меньшим номером.
     - Если такая есть, то мы увеличиваем счётчик num: $num = num + 1$.
     - Если такой нет, то меняем значение $IsEven$ на противоположное и увеличиваем счётчик num: $num = num + 1$.

   Заметим, что в таком алгоритме бит $IsEven$ меняет своё значение, только если текущая вершина не связана ребром ни с одной из ранее рассмотренных, то есть лежит в новой компоненте связности относительно найденных ранее алгоритмом. Таким образом, после окончания алгоритма $G \in EVENCOMM \Leftrightarrow IsEven = 1 \Rightarrow EVENCOMM \in \mathcal{L}$. +1

2. Для того, чтобы проверить принадлежность (G,e) к $EDGEUCYCLE$, где $G = (V,E)$ рассмотрим граф $\tilde{G} = (V,\tilde{E})$, где $\tilde{E}$ содержит все рёбра E, кроме e. Пусть $e = (u,v)$, тогда, очевидно, $(G,e) \in EDGEUCYCLE \Leftrightarrow (\tilde{G},u,v) \in UPATH$, что мы проверяем за логарифмическую память по теореме Рейнгольда, так что $EDGEUCYCLE \in \mathcal{L}$. +1

3. Сведём $XOR2SAT$ к языку $BIPARTITE$, который лежит в $\mathcal{L}$, доказательство этого есть в конспекте. По формуле φ будем строить граф $G = (V,E)$ следующим образом:

$V = \{x_i \mid$ переменная x_i лежит в формуле $\varphi\}$, а вершины x_i и x_j будем соединять ребром тогда и только тогда, когда в формуле φ есть конъюнкт вида $x_i \oplus x_j$. Таким образом мы можем за логарифм от длины входа построить наш граф G: перебором всех переменных заполняем матрицу смежности. Не вполне ясно, почему.

Докажем, что $\varphi \in XOR2SAT \Leftrightarrow G \in BIPARTITE$: Получилось д-во 1:1 как в обычной полиномиальной сводимости, если нет разницы, то это надо объяснить

- $\Rightarrow$: $\varphi \in XOR2SAT \Rightarrow \varphi$ выполнима, то есть существует такой набор переменных, что в каждом конъюнкте вида $x_i \oplus x_j$ справедливо, что x_i и x_j принимают различные значения, так что если в φ есть конъюнкты $x_i \oplus x_j$ и $x_i \oplus x_k$, то $x_j = x_k$, а следовательно $x_j \oplus x_k = 0$ и такого конъюнкта в φ нет, то есть в нашем графе G x_j и x_k ребром соединены не будут, а следовательно в нашем графе нет треугольников, а циклов большей нечётной длины в графе быть не может в принципе, так что граф двудолен.
- $\Leftarrow$: G двудольный $\Rightarrow$ примем все переменные одной доли равными 0, другой доли — равными 1, тогда в каждом конъюнкте вида $x_i \oplus x_j$ справедливо, что x_i и x_j принимают различные значения, так что φ выполняется на этом наборе.

Таким образом мы доказали, что $XOR2SAT \leqslant_L BIPARTITE \wedge BIPARTITE \in \mathcal{L} \Rightarrow XOR2SAT \in \mathcal{L}$, что и требовалось.

***Задача* 2.** Докажите, что $2SAT \in \mathcal{NL}$.

А куда переписать? А хватит места?

Решение. Во-первых вспомним, что $x_i \vee x_j \Leftrightarrow \overline{x_i} \to x_j$. А во-вторых, что $\mathcal{NL} = co - \mathcal{NL}$ по Immerman–Szelepcsényi theorem. Тогда мы можем каждый из наших дизъюнктов переписать в виде импликации. Таким образом, в качестве сертификата для нашей машины Тьюринга возьмём цепь импликаций вида $x_i \to \ldots \to \overline{x_i} \to \ldots \to x_i$, наша МТ будет идти по цепочке и смотреть, есть ли каждая импликация в нашей исходной формуле. Докажем, что если такая цепь есть в нашей исходной формуле, то она невыполнима: от противного: пусть $x_i = 1$, тогда все последующие литералы в импликации также должны быть равными 1 (так как исходная формула равняется 1 на данном наборе переменных), но тогда получим импликацию $1 \to \overline{x_i} = 1 \to 0 = 0$ — противоречие, если же $x_i = 0$, то аналогично рассматривая вторую половину цепочки получаем импликацию $1 \to x_i = 1 \to 0 = 0$ — противоречие.
Таким образом $2SAT \in co - \mathcal{NL} \Rightarrow 2SAT \in \mathcal{NL}$.

***Задача* 3.** Докажите, $\mathcal{NL}$-полноту языка $2SAT$.

Решение. Сведём к $2SAT$ $co - \mathcal{NL}$-полный язык $\overline{PATH}$. Так как $co - \mathcal{NL} = \mathcal{NL}$, то язык $\overline{PATH}$ также является $\mathcal{NL}$-полным языком. Построим КНФ следующим образом: (имеем тройку (G, s, t), как элемент $\overline{PATH}$). Заведём по одной переменной для каждой из вершин, внесём в нашу формулу дизъюнкты s и $\overline{t}$. А для $(x, y) \in E$ внесём в формулу дизъюнкт $\overline{x} \vee y$. Докажем теперь, что пути из s в t нет $\Leftrightarrow$ КНФ выполнима.

- $\Rightarrow$: возьмём все вершины, которые достигаются из вершины s и пометим соответствующие им переменные, как истинные, а все остальные, как ложные. $s = 1, t = 0$. Тогда в нашем графе нет рёбер, которые идут от вершины с соответствующей ей переменной, равной 1 к вершине с соответствующей ей переменной, равной 0. От противного: если есть ребро (u,v), где $u = 1, v = 0$, то существует путь $s \to \ldots \to u \to v$, то есть v достижима из s. Тогда все дизъюнкты вида $\overline{x} \vee y$, равно как s и $\overline{t}$ обращаются в 1, следовательно КНФ обращается в 1 на этом наборе $\Rightarrow$ выполнима.

- $\Leftarrow$: КНФ выполнима $\Rightarrow$ существует набор, на котором формула обращается в 1, возьмём его. Допустим, что в КНФ был дизъюнкт $\overline{x} \vee y$. Он обращается в 1, так что не может быть такого, чтобы x был равен 1, а y обращался в 0. Таким образом, можно построить достижимые из s вершины равно как в предыдущем пункте: все вершины, достижимые из s обращаются в 1 (так как если $(s, u) \in E$, то есть дизъюнкт $\overline{s} \vee u = 1$, где $s = 0 \Rightarrow u = 1$. Но так как КНФ выполнима, то $\overline{t} = 1$, то есть $t = 0$ — недостижимая из s вершина

Таким образом мы построили сводимость $\overline{PATH} \leqslant_L 2SAT$, следовательно $2SAT$ также является $\mathcal{NL}$-полным языком. Слова не понятно, почему это log-сводимость.
Поясните, тогда зачту последние 3.

Задача 4. Докажите, что класс $P/poly$ не изменится, если в качестве размера вместо числа вершин брать число рёбер.

Решение. $P/poly$ — класс языков, распознающихся семейством схем полиномиального размера (где размер есть число вершин в минимального схеме). Докажем, что если брать тут в качестве размера число рёбер, то класс не изменится:

- возьмём схему, распознающую язык и имеющую полиномиальный размер в смысле вершин: пусть этот размер есть $f(n) = poly(n)$. Так как всего в этой схеме рёбер $\leqslant f(n) \cdot f(n) = poly(n)$, то язык также распознаётся схемой с полиномиальным размером в смысле рёбер.

- возьмём схему, распознающую язык и имеющую полиномиальный размер в смысле рёбер: пусть этот размер есть $f(n) = poly(n)$. Так в нашей схеме $\leqslant n$ изолированных вершин (их не может быть больше, чем входных переменных), а остальных вершин не может быть больше общего числа рёбер, то всего вершин $\leqslant n + f(n) = poly(n)$. Таким образом этот язык также распознаётся схемой с полиномиальным размером в смысле вершин.

Таким образом мы доказали, что язык распознаётся семейством схем полиномиального размера в смысле вершин $\Leftrightarrow$ язык распознаётся семейством схем полиномиального размера в смысле рёбер. То есть эти классы действительно совпадают.

+1

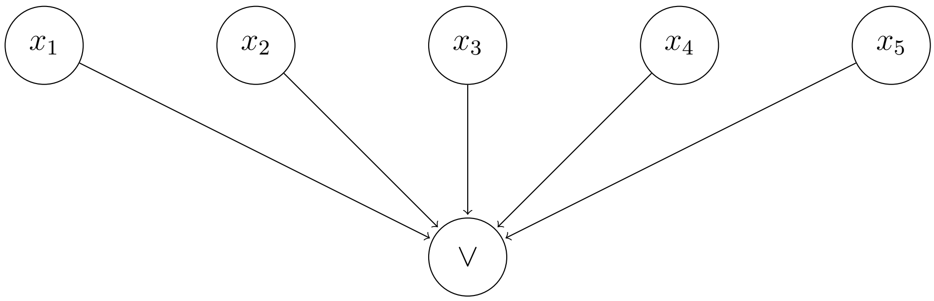
Задача 5. Докажите, что класс $P/poly$ не зависит от того, какая входящая степень разрешена для вершин типов $\wedge$ и $\vee$.

Решение. Пусть теперь разрешённая входящая степень есть N, обозначим получившийся класс как $P/poly - N$. Докажем, что на самом деле $P/poly = P/poly - N$:

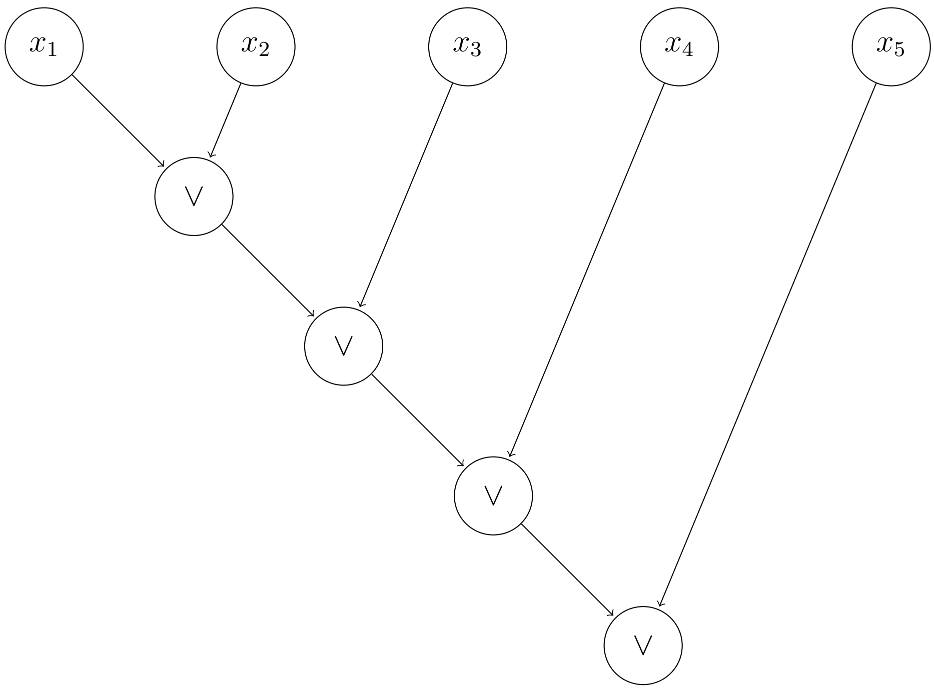
- $P/poly \subseteq P/poly - N$: верно, так как, очевидно, мы можем не обращать внимание на увеличение входящей степени и продолжать пользоваться схемами, где в каждую вершину вида $\wedge$ и $\vee$ входит не больше двух рёбер.

- $P/poly - N \subseteq P/poly$: развернём каждую из вершин вида $\wedge$ и $\vee$ «слева-направо», как показано ниже:

  Изначально имеем следующую схему:



  Разворачиваем «слева-направо»:



  Таким образом можно видеть, что если у нас была схема на m вершинах, причём $m = poly(n)$, то после такого эквивалентного преобразования мы получаем схему с $\leqslant N \cdot poly(m) \leqslant m \cdot poly(poly(n)) = poly(n)$ вершин. Таким образом размер остаётся полиномиальным, то есть $P/poly - N \subseteq P/poly$, чего нам и не хватало для доказательства того, что $P/poly = P/poly - N$.

+1