Задание №3 — «NP-completeness»
Лиана Хазалия, 3 курс 2 группа
2019/04/05, 13:25:48

Задача 99-1 [NP-Complectness]

Самый длинный путь (LongestPath)

Задан граф $G = (V, E)$ и положительное число $K \leq |V|$. Имеется ли в G простой путь (то есть путь, не проходящий дважды ни через одну вершину), состоящий не менее чем из K рёбер?

Решение.

Для решения предложенной задачи потребуется знание того, что задача о нахождении гамильтонова пути в неориентированном графе (UHamPath) является **NP**-полной.

Так как длина гамильтонова пути равна $|V| - 1$ (проходит по всем вершинам единожды, а также включает каждое из рёбер не более одного раза), то при нахождении гамильтонова пути решаем поставленную задачу для $K = |V| - 1$. То есть известная задача UHamPath суть частный случай предложенной. Следовательно, предложенная задача LongestPath **NP**-сложна.

Рассматриваемая задача принадлежит **NP**, так как для заданного графа $G = (V, E)$ сертификат имеет вид последовательности, состоящей из вершин множества $V' \subseteq V$. При верификации проверяется, что в эту последовательность каждая вершина из V' входит единожды и что каждая пара воследовательных вершин соединена ребром, то есть сертификат является действительным. Подобная проверка выполняется за полиномиальное время.

Из того, что задача LongestPath **NP**-сложна и что LongestPath $\in$ **NP**, следует её **NP**-полнота.

Задача 100-10 [NP-Complectness]

Неэквивалентность регулярных выражений, не содержащих звёздочек (Star-free RegEx Ineq)

Заданы два не содержащие звёздочек регулярных выражения E_1 и E_2 в конечном алфавите Σ. Такое выражение определяется следующим образом:

1. *любой символ σ алфавита Σ есть не содержащее звёздочек регулярное выражение,*
2. *если e_1 и e_2 — два не содержащие звёздочек регулярных выражения, то и слова e_1e_2 и $(e_1 \vee e_2)$ также не содержащие звёздочек регулярные выражения.*

Верно ли, что E_1 и E_2 представляют различные языки в алфавите Σ? (Язык, представляемый символом $\sigma \in \Sigma$, если $\{\sigma\}$, а если e_1 и e_2 представляют соответственно языки L_1 и L_2, то e_1e_2 представляет язык $\{xy : x \in L_1, y \in L_2\}$, а $(e_1 \vee e_2)$ представляет язык $L_1 \cup L_2$.)

Решение.

Докажем, что задача лежит в классе **NP**. Чтобы убедиться в неэквивалентности двух регулярных выражений, не содержащих звёздочек, должно быть найдено слово x, которое $x \in L_1$ и $x \notin L_2$ или наоборот. Без нарушения общности, рассмотрим первый случай.

Нужно проверить, что $x \in L_1$ и что $x \notin L_2$. Оба этих факта можно проверить сравнением слова x с деревьями, которые соответствует регулярным выражениям E_1 и E_2, порождающим языки L_1 и L_2 соответственно.

Дерево регулярного выражения — бинарное дерево, внутренним вершинам которого присвоена одна из операций дизъюнкции или конъюнкции, а вершинам-листьям присвоены символы алфавита $\Sigma^* = \Sigma \cup \{\emptyset\}$. Обходом дерева в обратном порядке можем сравнить слово x с листьями каждого поддерева и узнать, какие префиксы слова x каким поддеревьям могут соответствовать. По окончанию обхода дерева можно получить ответ: принадлежит слово x языку, порожденному регулярным выражением, обход дерева которого был произведён, или нет.

Для представления регулярного выражения как бинарного дерева, а также для его обхода в обратном порядке, требуется линейное время. Предположим, что слово x имеет длину n, тогда у x можно различать $n+1$ префикс. Следовательно, асимпотика алгоритма установления принадлежности слова x языку L_1 равна $\mathcal{O}\left(n \cdot max\left(|E_1|, |E_2|\right)\right)$. В данном случае важную роль играет то, что регулярные выражения E_1 и E_2 не содержат звёздочек Клини, что позволяет говорить о том, что высота дерева ограничена длиной регулярного выражения, ему соответствующего: в листьях символы алфавита Σ^*, дизъюнкция не увеличивает длину строки, а конъюнкция складывает длины, — длина регулярного выражения $n \leq max\left(|E_1|, |E_2|\right)$ и верификация выполняется за полиномиальное время.

Теперь рассмотрим задачу 3-sat: дан набор $C = \{C_i\}_{i=1}^{m}$ дизъюнктов на конечном множесте переменных $U = \{x_i\}_{i=1}^{n}$ таких, что $|C_i| = 3$, $1 \leq i \leq m$. Существует ли на U набор значений истинности, при котором выполняются все дизъюнкции из C?

Докажем, что 3-sat $\leq_P$ Star-free RegEx Ineq. Пусть дан экземпляр ϕ задачи 3-sat с литералами $x_1, x_2, \ldots, x_n$. Построим по экземпляру задачи 3-sat экземпляр требуемой следующим образом: так как каждый литерал может принимать значение 0 или 1, то зададим алфавит как $\Sigma = \{0, 1\}$. Далее, пусть регулярное выражение E_1 порождает все возможные присваивания значений литералам $x_1, \ldots, x_n$, условно множество всех возможных слов длины n над алфавитом $\{0, 1\}$, то есть мощность его 2^n и E_1 имеет вид:

$$E_1 = \underbrace{(0 \vee 1) \cdot (0 \vee 1) \cdot \ldots \cdot (0 \vee 1)}_{n}.$$

Регулярное выражение E_2 построим по дизъюнктам так, чтобы оно порождало язык, множество слов которого соответствовует присваиванию значений литералам, при которых исходная конъюнкция дизъюнктов принимает ложное значение. Для того чтобы исходная конъюнкция дизъюнктов принимала ложное значение достаточно, чтобы i-ый дизъюнкт принимал ложное значение. Язык, соответствующий присваиванию значений для случая, когда $C_i = 0$, суть множество слов, для которых верно следующее: если в i-ом дизъюнкте присутствует литерал x_j, то на j-ой позиции в слове должен быть 0, если в i-ом дизъюнкте присутствует отрицание литерала x_j, то на j-ой позиции в слове должна быть 1, если в i-ом дизъюнкте C_i отсутствует литерал x_j, то на j-ой позиции в слове может быть как 0, так и 1. Регулярное выражение R_i $(i = \overline{1, m})$, порождающее описанный язык, выглядит как конъюнкция n выражений r_j:

$$r_j = \begin{cases} 0 & x_j \in C_i, \\ 1 & \overline{x_j} \in C_i, \\ (0 \vee 1) & x_j \notin C_i, \end{cases}$$

где $j = \overline{1, n}$.

Например, для дизъюнкта $(x_2 \vee \bar{x_3} \vee x_5)$ при $n = 5$ регулярное выражение будет иметь вид $(0 \vee 1) \cdot 0 \cdot 1 \cdot (0 \vee 1) \cdot 0$.

Так, E_2 — дизъюнкция m регулярных выражений R_i, каждое из которых порождает множество слов, для которых $C_i = 0$, построение каждого из которых описано выше и может быть проведено в течение полиномиального времени.

Теперь докажем, что удовлетворяющее присваивание значений литералов исходного экземпляра ϕ задачи 3-sat существует тогда и только тогда, когда языки L_1 и L_2 не равны в алфавите $\Sigma = \{0, 1\}$.

По присвоенным литералам значениям построена строка, соответствующая проведенному присваиванию: длина строки равна n, а i-ый символ равен 1 или 0, если i-ый литерал равен 1 или 0 соответственно. Очевидно, что такая строка лежит в L_1, так как L_1 содержит слова, соответствующие всем возможным присваиваниям значений литералам $x_1, \ldots, x_n$ выражения ϕ. Но так как все дизъюнкты, ввиду истинности ϕ, истины, слово не будет принадлежать L_2.

В обратную сторону, если слово $x \in L_1$ и $x \notin L_2$, то соответствующее слову x призваивание значений литералам $x_1, \ldots, x_n$ даёт истиное значение выражения ϕ. Если значение ϕ ложно, то хотя бы один из дизъюнктов будет принимать значение 0, а, следовательно, слово x лежит в E_2. Противоречие.

Хотелось бы дополнить вышесказанное: язык L_1 содержит все возможные слова длины n над алфавитом $\Sigma = \{0, 1\}$, а язык L_2 содержит только те слова длины n над алфавитом $\Sigma = \{0, 1\}$, которые соответствуют присваиванию значений литералам, при котором $\phi = 0$, то есть $L_2 \subseteq L_1$. И равенство достигается, только если все возможные присваивания значений литералам $x_1, \ldots, x_n$ дают равенство $\phi = 0$, то есть в случае, когда $\phi \equiv 0$.

Таким образом, доказали, что данная задача Star-free RegEx Ineq принадлежит в **NP** и 3-sat $\leq_P$ Star-free RegEx Ineq. Следовательно, Star-free RegEx Ineq является **NP**-полной.

Задача 7 [NP-ISSUES]

Так как задача о трёхмерном сочетании является NP-полной, естественно ожидать, что аналогичная задача о четырёхмерном сочетании будет хотя бы не менее сложной. Определим четырехмерное сочетание следующим образом: для заданных множеств W, X, Y и Z, каждое из которых имеет размер n, и набора C упорядоченных четверок в форме (w_i, x_j, y_k, z_l) существуют ли n четверок из C, среди которых никакие два не имеют общих элементов?
Докажите, что задача о четырехмерном сочетании является NP-полной.

Решение.

Покажем, что предложенная задача 4-dim matching **NP**-сложна, доказав 3-dim matching $\leq_P$ 4-dim matching. Сведение начинается с экземпляра задачи 3-dim matching. Пусть дано множество $M_3 \subseteq W \times X \times Y$, где W, X и Y — непересекающиеся множества, содержащие одинаковое число элементов q. Четвертое множество Z мощности q берется так, чтобы пересечение с любым из уже имеющихся было пусто. Задаём некоторое биективное соответствие между множествами Y и Z. Каждая четверка — существовавшая ранее тройка, к которой добавлена четвёртая координата, которой присвоено значение элемента множества Z, который соответствует элементу множества Y в третьей координате взятой тройки в ранее заданной биекции.

Покажем, что проведенное преобразование является сведением. Допустим, что для множества четвёрок $M_4 \subseteq W \cup X \cup Y \cup Z$ существует такое подмножество $M_4' \subseteq M_4$, что $|M_4'| = q$ и что любой элемент множества $W \cup X \cup Y \cup Z$ принадлежит ровно одной из четверок множества M_4'. Тогда верно и утверждение, что любой элемент множества $W \cup X \cup Y$ принадлежит ровно одной из троек множества $M_3' \subseteq M_3$, которое получено удалением четвертой координаты в каждой четверке. Проведём и обратные рассуждения. Предположим, что нет решения у задачи 4-dim matching. Тогда как минимум один из элементов $W \cup X \cup Y$ не покрыт или покрыт с повторами, так как не покрыть (или покрыть с повторами) элементы множества Z значит не покрыть (или покрыть с повторами) и элементы множества Y, потому что между ними имеется биекция.

Чтобы показать, что задача 4-dim matching лежит в **NP**, воспользуемся множеством $M_4 \subseteq W \cup X \cup Y \cup Z$ в качестве сертификата. Проверить, лежит ли каждый элемент из множества $W \cup X \cup Y \cup Z$ в ровно одной из четверок множества M_4 можно в течение полиномиального времени.

Задача 16 [NP-ISSUES]

Рассмотрим задачу характеристики множества по размерам его пересечений с другими множествами. Имеется конечное множество U размера n, а также набор $A_1, \ldots, A_m$ подмножеств U. Также заданы числа $c_1, \ldots, c_m$. Вопрос звучит так: существует ли такое множество $T \subset U$, что для всех $i = 1, 2, \ldots, m$ мощность $T \cap A_i$ равна c_i? Назовем его задачей выведения пересечений с входными данными U, $\{A_i\}$ и $\{c_i\}$.
Докажите, что задача выведения пересечений является NP-полной.

Решение.

Рассмотрим задачу 3-dim Mathing, **NP**-полнота которой известна: дано множество $M \subseteq W \times X \times Y$, где W, X и Y — непересекающиеся множества, содержащие одинаковое число элементов q. Верно ли, что M содержит трёхмерное сочетание, то есть подмножество $M' \subseteq M$, такое, что $|M'| = q$ и никакие два разных элемента M' не имеют ни одной равной координаты?

Сначала докажем, что задача **NP**-сложна по причине сводимости к ней 3-dim Matching. Пусть дан экземпляр задачи 3-dim Matching: некоторые непересекающиеся множества $W = \{w_i\}_{i=1}^q$, $X = \{x_i\}_{i=1}^q$, $Y = \{y_i\}_{i=1}^q$ мощности $q \in \mathbb{N}$ и некоторое множество $M \subseteq W \times X \times Y$. Пусть множество U исходной задачи совпадает с множеством троек M. Далее, зададим множества подмножеств $\{A_i\}_{i=1}^{3n}$ множества U следующим образом:

$$\{A_i\}_{i=1}^q = \{A_i = \{(w_i, x, y) | x \in X,\, y \in Y\}\}_{i=1}^q\,,$$
$$\{A_{q+i}\}_{i=1}^q = \{A_{q+i} = \{(w, x_i, y) | w \in W,\, y \in Y\}\}_{i=1}^q\,,$$
$$\{A_{2q+i}\}_{i=1}^q = \{A_{2q+i} = \{(w, x, y_i) | w \in W,\, x \in X\}\}_{i=1}^q\,.$$

То есть в подмножество, например, A_1 входят все тройки, у которых в первой координате значение равно значению элемента w_1 множества W. Также определим ещё одно подмножество $A_{3q+1} = U$. Множество параметров $\{c_i\}_{i=1}^{3q+1}$ зададим как $c_i = 1\ \forall\, i = \overline{1, 3q}$, а $c_{3q+1} = q$. Также сто́ит отметить, что для создания экземпляра рассматриваемой задачи из экземпляра известной (3-dim matching) требуется время $\mathcal{O}(q^3)$.

Такое преобразование является сведением, так как если множество $T \subseteq U$ для сформулированной задачи выведения пересечений с описанными ранее входными данными U, $\{A_i\}$ и $\{c_i\}$ существует, то взято ровно q троек, так как $c_{3q+1} = |A_{3q+1} \cap T| = |U \cap T| = |M \cap T| = q$. В свою очередь, каждый элемент из множества $W \cup T \cup Y$ в T встречается единожды, в силу того что мощность пересечения $|T \cap A_i| = |T \cap \{\text{все тройки с фиксированным элементом}\}| = c_i = 1$ для любого $i = \overline{1, 3q}$. То есть множество T является искомым множеством M' для задачи 3-dim matching. Если же решения сформулированной задачи не существует, то по крайней мере одно ограничение не было выполнено, следовательно, по крайней мере один элемент из $W \cup X \cup Y$ не был покрыт или какой-то учтен несколько раз. Таким образом, сведение верно.

Чтобы доказать принадлежность поставленной задачи классу **NP**, необходимо доказать, что существует верификатор такой, что при наличии действительного сертификата его можно проверить за полиномиальное время. Пусть имеется верификатор, который для входа $(U, \{A_i\}, \{c_i\})$ проверяет, что некоторое множество T (сертификат) является подмножеством множества U и что для каждого подмножества $A_i \in A$ выполнено $|A_i \cap T| = c_i$. Все перечисленные операции могут быть проверены за время, полиномиально зависящее от входных параметров. Отсюда имеем, что рассматриваемая задача выведения пересечений лежит в **NP**.

Таким образом, исходная задача **NP**-полна, потому что была доказана возможность полиномиального сведения известной **NP**-полной задачи к рассматриваемой, лежащей в **NP**.

Электронная версия документа доступна в репозитории GitHub
LianaKhazaliya/Algorithms/Task3