

Лекция 11. Примеры NP-полных задач

Для того чтобы доказать NP-полноту задачи, необходимо, согласно определению, доказать, во-первых, принадлежность задачи классу NP, а во-вторых, доказать возможность построения алгоритма сведения любой задачи класса NP к данной. Очевидно, что это достаточно сложная процедура.

Лемма 3, доказанная выше, дает реальный способ доказательства NP-полноты произвольной задачи L , заключающийся в следующем:

- 1) доказать, что $L \in NP$;
- 2) выбрать некоторую NP-полную задачу L' ($L' \in NPC$) и свести ее к данной;
- 3) доказать, что построенный алгоритм сведения полиномиален.

Однако чтобы воспользоваться этим способом, необходимо иметь в арсенале хотя бы одну задачу, NP-полнота которой уже доказана. Роль такой NP-полной задачи принадлежит задаче распознавания из булевой логики, которую обычно называют ВЫПОЛНИМОСТЬ (сокращенно ВЬП).

Задача ВЫПОЛНИМОСТЬ

Пусть $X = \{x_1, x_2, \dots, x_m\}$ – множество булевских переменных. На множестве переменных построено множество литералов $U = \{u_1, u_2, \dots, u_{2 \cdot m}\}$. Литералом и называется переменная x или ее отрицание \bar{x} .

На множестве литералов построена формула φ , являющаяся конъюнкцией произвольного числа дизъюнкций, состоящих из произвольного числа литералов. Формула φ называется *выполнимой* в том и только в том случае, если найдется некоторый набор значений переменных на множестве X , на котором формула φ принимает истинное значение. Такой набор значений переменных называется *выполняющим набором* для формулы φ . В противном случае (если нет выполняющего набора значений переменных) формула называется *невыполнимой*.

Задача ВЫПОЛНИМОСТЬ формулируется следующим образом.

Дано множество переменных X , на котором построена формула φ . Определить, выполнима формула φ или нет.

Например, пусть $X = \{x_1, x_2\}$, $\varphi = (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1) \wedge (x_1 \vee x_2)$. Это индивидуальная задача из ВЫПОЛНИМОСТИ, ответ на которую – «да», так как существует выполняющий набор $x_1 = \text{false}$, $x_2 = \text{true}$. А формула $\varphi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1) \wedge (x_1 \vee x_2)$ – индивидуальная задача, ответ на которую – «нет», так как не существует выполняющего набора.

NP-полнота данной задачи была доказана Куком.

Теорема Кука. Задача ВЫПОЛНИМОСТЬ NP-полна.

Для доказательства использовался аппарат машин Тьюринга (см. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи).

Задача 3-ВЫПОЛНИМОСТЬ

Формулировка задачи 3-ВЫПОЛНИМОСТЬ (3-ВЫП): дано множество переменных X , на котором построена формула φ , являющаяся конъюнкцией произвольного числа трехлитеральных дизъюнкций. Определить, выполнима формула φ или нет.

Например, пусть $X=\{x_1, x_2, x_3\}$, $\varphi=(\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$. Формула выполнима, так как существует выполняющий набор, например, $x_1=false$, $x_2=true$, $x_3=false$.

Теорема. Задача 3-ВЫП NP -полна ($3\text{-ВЫП} \in NPC$).

Доказательство

1. Задача $3\text{-ВЫП} \in NP$, так как проверяется за полиномиальное время (в качестве сертификата можно рассматривать выполняющий набор). Действительно, для того чтобы проверить, является ли некоторый набор выполняющим для формулы из k дизъюнкций, необходимо выполнить действий $O(3 \cdot k)$.

2. Построим алгоритм сведения NP -полной задачи ВЫП к задаче 3-ВЫП, т. е. преобразования произвольной формулы $\varphi_{\text{вып}} \in \text{ВЫП}$ к задаче $\varphi_{3\text{-вып}} \in 3\text{-ВЫП}$. Для этого укажем правила преобразования каждой дизъюнкции C_j формулы $\varphi_{\text{вып}}$ в конъюнкцию трехлитеральных дизъюнкций C'_j формулы $\varphi_{3\text{-вып}}$ так, чтобы выполнялось условие $C_j - \text{выполнима} \Leftrightarrow C'_j - \text{выполнима}$.

Пусть дизъюнкция C_j имеет вид: $C'_j = u_1 \vee u_2 \vee \dots \vee u_k$.

1) $k=1$, $C_j = u_1$.

Введем набор дополнительных переменных $Y=\{y_1, y_2\}$ и построим формулу C'_j следующим образом:

$$C'_j = (u_1 \vee y_1 \vee y_2) \wedge (u_1 \vee \overline{y_1} \vee y_2) \wedge (u_1 \vee y_1 \vee \overline{y_2}) \wedge (u_1 \vee \overline{y_1} \vee \overline{y_2}).$$

Очевидно, что истинность формулы C'_j зависит лишь от истинности литерала u_1 , следовательно, $C_j - \text{выполнима} \Leftrightarrow C'_j - \text{выполнима}$.

2) $k=2$, $C_j = u_1 \vee u_2$.

Введем набор дополнительных переменных $Y=\{y_1\}$ и построим формулу C'_j следующим образом:

$$C'_j = (u_1 \vee u_2 \vee y_1) \wedge (u_1 \vee u_2 \vee \overline{y_1}).$$

Очевидно, что истинность формулы C'_j зависит лишь от истинности выражения $u_1 \vee u_2$, следовательно, $C_j - \text{выполнима} \Leftrightarrow C'_j - \text{выполнима}$.

3) $k=3$, $C_j = u_1 \vee u_2 \vee u_3$.

Преобразования дизъюнкции не требуется, т. е. $C_j' = C_j$.

4) $k \geq 4$, $C_j = u_1 \vee u_2 \vee \dots \vee u_k$.

Введем набор дополнительных переменных $Y = \{y_1, y_2, \dots, y_{k-3}\}$ и построим формулу C_j' следующим образом:

$$C_j' = (u_1 \vee u_2 \vee y_1) \wedge \left\{ \bigcup_{i=1}^{k-4} (\overline{y_i} \vee u_{i+2} \vee y_{i+1}) \right\} \wedge (\overline{y_{k-3}} \vee u_{k-1} \vee u_k).$$

Например,

а) пусть $C_j = u_1 \vee u_2 \vee u_3 \vee u_4$, тогда $Y = \{y_1\}$ $C_j' = (u_1 \vee u_2 \vee y_1) \wedge (\overline{y_1} \vee u_3 \vee u_4)$;

б) пусть $C_j = u_1 \vee u_2 \vee u_3 \vee u_4 \vee u_5 \vee u_6$, тогда $Y = \{y_1, y_2, y_3\}$
 $C_j' = (u_1 \vee u_2 \vee y_1) \wedge (\overline{y_1} \vee u_3 \vee y_2) \wedge (\overline{y_2} \vee u_4 \vee y_3) \wedge (\overline{y_3} \vee u_5 \vee u_6)$.

Очевидно, что если формула C_j невыполнима, то C_j' также невыполнима, следовательно, C_j – выполнима $\Leftrightarrow C_j'$ – выполнима.

3. Построенный алгоритм является алгоритмом сведения, так как $\varphi_{\text{вып}}$ выполнима $\Leftrightarrow \varphi_{3\text{-вып}}$ выполнима. Данный алгоритм полиномиален, время преобразования пропорционально длине формулы $\varphi_{\text{вып}}$, а число трехлитеральных дизъюнкции в $\varphi_{3\text{-вып}}$ ограничено полиномом от $m \cdot n$, где m – число дизъюнкции в $\varphi_{\text{вып}}$, n – число литералов в $\varphi_{\text{вып}}$.

Таким образом, в силу леммы 3 задача 3-ВЫП NP -полна.

Задача о КЛИКЕ

Определение. Кликкой в неориентированном графе $G=(V, E)$ называется подмножество вершин $V' \subseteq V$, каждые две из которых соединены ребром графа.

Таким образом, клика – это множество вершин полного подграфа графа.

Оптимизационная задача о клике. Определить максимальный размер клики в данном графе.

Задача разрешения (КЛИКА). Даны граф G и число k . Требуется установить, есть ли в графе G клика размера k .

Для решения задачи можно перебрать все подмножества вершин размера k в графе G (их количество равно числу сочетаний – C_V^k) и проверить, есть ли среди них клика (сложность проверки для каждого подмножества $O(k^2)$). Таким образом, для решения задачи о клике требуется действий $O(k^2 \cdot C_V^k)$, т. е. это задача экспоненциальной сложности (труднорешаемая).

Теорема. Задача о клике NP -полна (КЛИКА $\in NPC$).

Доказательство.

1. Задача КЛИКА $\in NP$, так как проверяется за полиномиальное время (в качестве сертификата можно рассматривать клику размера k). Действительно, для того чтобы проверить, является ли некоторое подмножество из k вершин графа G кликой, необходимо выполнить действий $O(k^2)$.

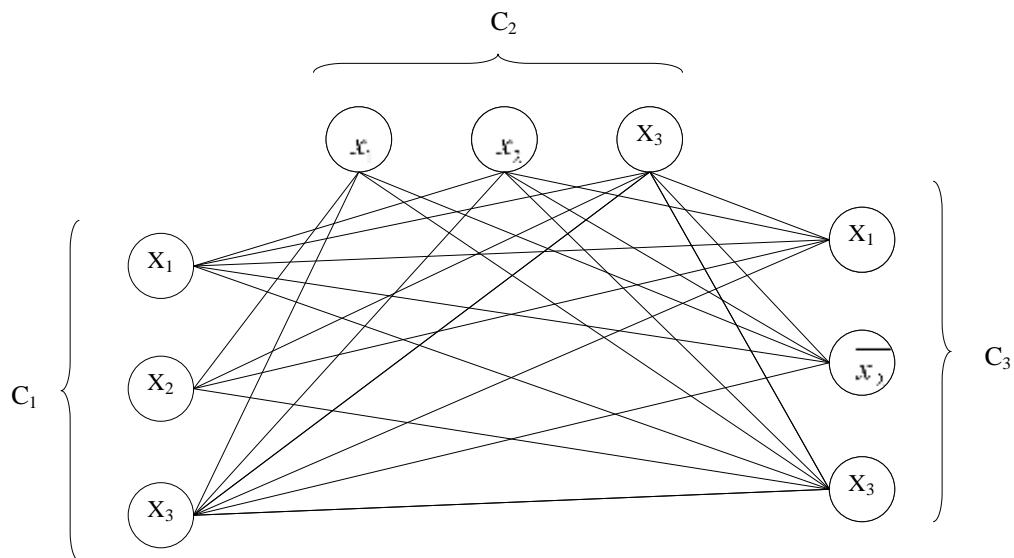
2. Построим алгоритм сведения NP -полной задачи 3-ВЫП к задаче КЛИКА, т. е. алгоритм преобразования произвольной формулы $\varphi_{3-вып} \in 3\text{-ВЫП}$ в пару $(G, k) \in \text{КЛИКА}$. Для этого укажем правила преобразования формулы $\varphi_{3-вып} = C_1 \wedge C_2 \wedge \dots \wedge C_k$ в граф $G=(V, E)$, в котором существует клика размера k тогда и только тогда, когда формула $\varphi_{3-вып}$ выполнима.

Пусть дана формула $\varphi_{3-вып} = C_1 \wedge C_2 \wedge \dots \wedge C_k$, где C_i – дизъюнкция трех литералов. Для построения графа G необходимо, во-первых, задать множество вершин V , а во-вторых, множество ребер E . Сделаем это следующим образом:

1) для каждой дизъюнкции строим по три вершины – по одной на каждый литерал, таким образом, всего в графе $3 \cdot k$ вершин;

2) построим ребра по следующему правилу: две вершины графа G соединяются ребром тогда и только тогда, когда они соответствуют литералам разных дизъюнкций и соответствующие им литералы совместны (т. е. один не является отрицанием другого).

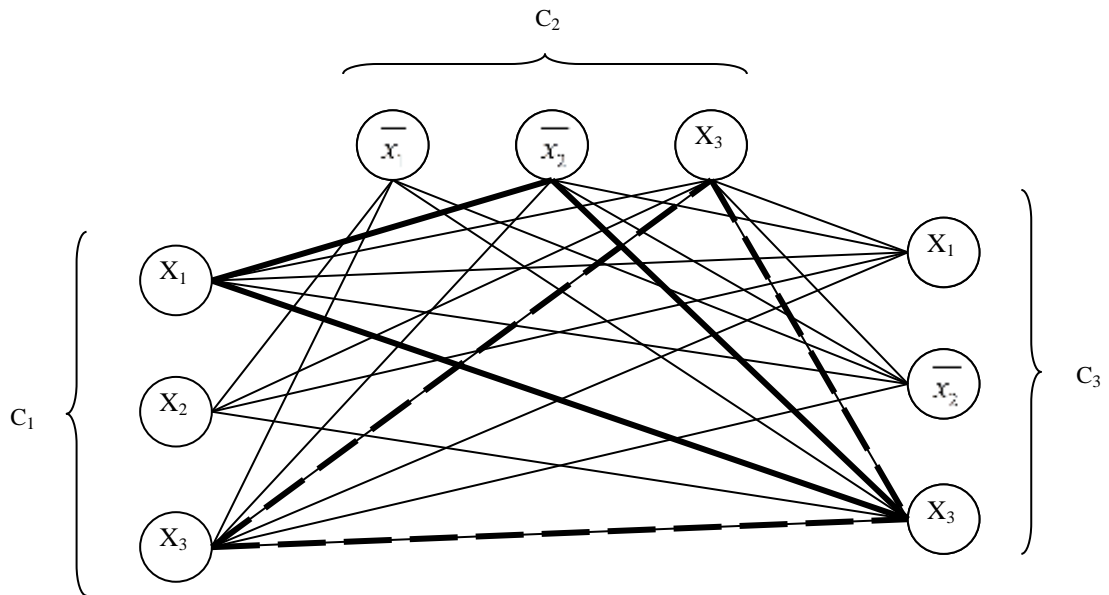
Например, дана формула $\varphi_{3-вып} = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3)$, тогда граф G будет иметь вид:



Покажем, что описанное преобразование действительно является сведением, т. е. докажем, что формула $\varphi_{3-вып}$ выполнима \Leftrightarrow в графе $G=(V, E)$ существует клика размера k .

1) Предположим, что формула имеет выполняющий набор, следовательно, в любой дизъюнкции формулы имеется хотя бы один истинный литерал. Выберем по одному истинному литералу из каждой дизъюнкции, так как они попарно совместны, то существуют ребра, их соединяющие, т. е. образуют клику.

Например, для формулы $\varphi_{3-вып} = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$ существует выполняющий набор $(x_1=true, x_2=false, x_3=true)$. Вершины, соответствующие этим литералам, попарно соединены ребрами, следовательно, образуют клику, которая выделена жирными линиями:



2) Пусть в графе есть клика размера k . В каждой тройке, соответствующей одной дизъюнкции, вершины не соединены друг с другом, следовательно, клика содержит ровно по одной вершине из каждой тройки. Рассмотрим соответствующие им литералы и объявим их истинными. Совместность литералов гарантирует, что их не придется объявлять одновременно истинными и ложными. Если не все переменные получили значения, то им значение придается произвольно, так как от них истинность формулы не зависит.

Например, в графе G есть клика, образованная вершинами (x_3, x_3, x_3) (клика выделена пунктиром). Объявляем литерал x_3 истинным, т. е. $x_3=true$. Действительно, эта переменная, независимо от значений, обращает формулу $\varphi_{3-вып} = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$ в истину, значит, значения для переменных x_1 и x_2 можно указать произвольно.

3) Построенный алгоритм является алгоритмом сведения, так как формула $\varphi_{3-вып}$ выполнима \Leftrightarrow в графе $G=(V, E)$ существует клика размера k . Данный алгоритм полиномиален, так как время сведения ограничено сверху полиномом $O((3 \cdot k)^2)$.

Таким образом, в силу леммы 3 задача КЛИКА NP -полна.