ЛЕКЦИЯ 3

СЛОЖНОСТЬ ВЫЧИСЛЕНИЙ

1. Задача о три-раскрасе

$$\text{3-COL} = \{G : \textit{вершины } G \textit{ можно раскрасить в } 3 \textit{ цвета}\},$$

где G — граф.

Граф раскрашивается так, чтобы не были соединены вершины одного цвета. Очевидно, это лежит в NP: если имеется раскраска, то можно проверить, что она правильная — все рёбра соединяют вершины разных цветов.

Сведём 3-SAT(3-КНФ) к этой задаче, то есть докажем, что задача является NP-трудной:

$$\text{3-SAT} \leqslant_p \text{3-COL}.$$

Док-во: Нарисуем палитру и проименуем эти три цвета следующим образом:

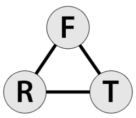

Рис. 3.1: Палитра

где T — истинный, F — ложный, R — красный.

Теперь построим более сложный граф:

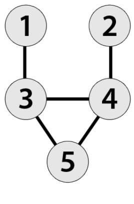

Рис. 3.2: Две вершины

Если вершины 1 и 2 одинаково раскрашены, то вершина 5 принимает такой же цвет. Если по-разному, то вершина 5 может принять цвет T (в этом случае может быть и другая раскраска, но важна именно такая).

Проиллюстрируем сказанное выше на рисунке **??**.

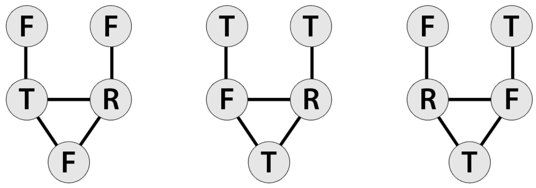

Рис. 3.3

То есть, внизу может быть T, если вверху все T или одновременно и T и F. Если же вверху все F, то и внизу стоит F. В этом смысле эта конструкция схожа с дизъюнкцией.

Теперь можем собрать тоже самое для трёх вершин:

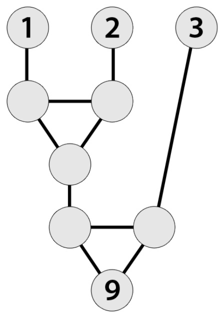

Рис. 3.4: Три вершины

Здесь верно аналогичное утверждение для трёх вершин, что если среди них хотя бы одна T, то можно раскрасить так, чтобы внизу была T. А если вверху все F, то внизу обязательно F.

Теперь соберём из этого 3-SAT.

Для каждой переменной рисуем гаджет-гантельку (далее — гантелька), концы которой соединим линиями красного цвета с R в палитре. Смысл в том, что если раскраска правильная, то гантельки раскрашены только в F или T, но не могут быть в R. То, как именно раскрашены гантельки, задаёт значение переменной: если левый конец окрашен в T, то переменная истинная, если правый, то переменная ложная. Гантелька не может иметь одинаково раскрашенные концы.

Пусть формула начинается с $(p_1 \vee \bar{p_2} \vee p_3)$. Для каждого дизъюнкта приделываем гаджет как на рисунке 3.4, причём если литерал переменной без отрицания, то нужно подсоединять к левому концу, а если с отрицанием, то к правому. Тогда картинка с соответствующими гаджетами будет выглядеть как на рисунке 3.5.

Так как внизу на гаджетах должно получиться T, то соединяем этот конец с F и R на палитре. И так далее. Для каждой скобочки рисуется такой гаджет.

Теперь, когда конструкция описана, нужно доказать, что выполняющий набор существует тогда и только тогда, когда существует правильная раскраска в три цвета.

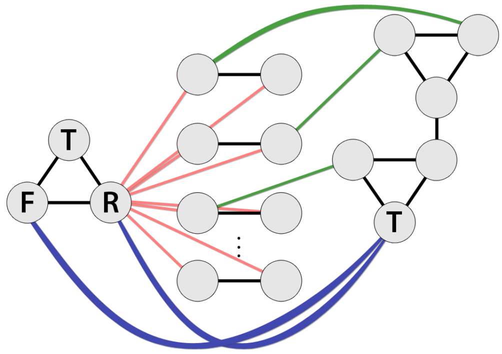

Рис. 3.5: 3-SAT для $(p_1 \vee \bar{p_2} \vee p_3)$

Пусть есть выполняющий набор. Тогда если соответствующая переменная выполняющего набора истинная, то левый конец соответствующей гантельки покрасим в T, если ложная — в F, а правый красим в противоположный цвет (в F или T соответственно). Палитра имеет фиксированную раскраску. Поскольку имеется выполняющий набор, тогда в каждой скобочке есть хотя бы один истинный литерал, а это значит, что к каждому гаджету, который построен по скобочке, подсоединена хотя бы одна вершина, которая покрашена в истинный цвет. Тогда существует раскраска этого гаджета, где внизу тоже истинный цвет. Значит, нет противоречия с тем, что этот конец связан с F и R на палитре.

Теперь докажем обратное: если существует правильная раскраска в три цвета, то существует выполняющий набор.

Так как существует раскраска, то внизу гаджет точно принимает значение T, а гантельки, соответственно, раскрашены только в T или F. Это значит, что при правильной раскраске в подсоединённых концах гантелек (входах) не могут быть одновременно все F, хотя бы один вход окрашен в T. А это значит, что в соответствующей дизъюнкции есть хотя бы один истинный литерал. Таким образом, в каждом дизъюнкте есть истинный литерал, причём формула на этом наборе истинная, что означает наличие выполняющего набора. ■

2. Существование гамильтонова пути в ориентированном графе

DHAMPATH $= \{(G, s, t)$: в ориентированном графе G существует ориентированный гамильтонов путь из s в $t\}$.

Ориентированный гамильтонов путь — это ориентированный путь, который проходит через все вершины. Покажем, что 3-SAT сводится к этой задаче:

$$\text{3-SAT} \leqslant_p \text{DHAMPATH}.$$

Нарисуем гаджет для переменных (см. рисунок 3.6). Этот гаджет состоит из ромбов, каждый из которых соответствует переменной, по которым можно двигаться сверху вниз по траектории вида $s \to 1 \to 3$ (зиг-загом) или $s \to 2 \to 3$ (заг-зигом) до t. Ещё нужно

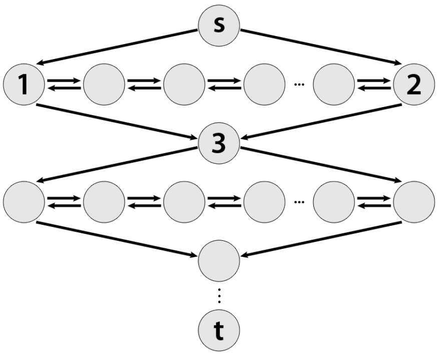

Рис. 3.6: Гаджет для переменных

указать, сколько вершин заключено между элементами типа 2 и 3. Их должно быть полиномиальное число (утроенное число скобок + 1).

Каждому дизъюнкту в графе будет соответствовать отдельная вершина. Вершине e соответствует $(p_1 \vee \bar{p_2} \vee p_3)$ (см. рисунок 3.7). После вершины 1 на гаджете оставляем промежуточную вершину (левую). Дальше, если переменная входит без отрицания, то из средней вершины идём в e, а в правую вершину входим из e (рассматриваем первую тройку вершин после вершины 1). Если переменная входит с отрицанием, то меняем направления входа и выхода из e. В данном примере вершина e соединяется с тремя ромбами, так как в скобку входят три переменные.

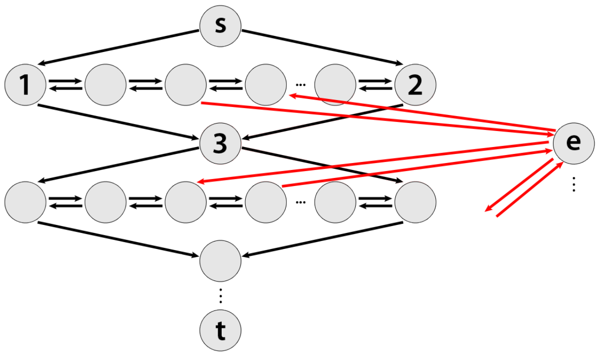

Рис. 3.7: Вершина e соответствует $(p_1 \vee \bar{p_2} \vee p_3)$

Теперь, когда конструкция описана, докажем, что имеет место сводимость.

Док-во: Докажем, что есть гамильтонов путь.

Пусть есть выполняющий набор. Это значит, что в каждой скобке есть один истинный литерал. Соответственно, если переменная истинная, будем проходить по левой части зиг-загом, а если ложная, то заг-зигом. Таким образом, все вершины слева будут автоматически пройдены, а все вершины справа также будут пройдены, так как каждая из них согласована с направлением прохода хотя бы одной из входящих переменных, и её можно посетить в соответствующий момент.

Докажем обратное.

Пусть есть гамильтонов путь. Если путь регулярный, то выполняющий набор по нему строится достаточно просто. Каждый из ромбов обходим либо зиг-загом (тогда говорим, что соответствующая переменная истинная), либо заг-зигом (тогда — ложная), и это выполняющий набор. Так как каждая из точек была пройдена, если ромб обходили зиг-загом, то в скобочку входит истинная переменная (значит, скобочка истинная), а если заг-зигом — ложная переменная, и в скобочку входит отрицание (скобочка тоже истинная). Таким образом, если есть регулярный путь, то тогда все скобочки истинные при таком наборе значений, и, соответственно, формула выполнимая.

Докажем, что нерегулярного гамильтонова пути быть не может.

Нерегулярный гамильтонов путь — это путь, в котором при переходе из одного ромба к вершинам справа можно вернуться в другой ромб, а не в тот же. В этом случае обязательно какая-то вершина не будет посещена, или какая-то вершина будет посещена дважды. Для этой цели были оставлены промежуточные вершины.

Идя заг-зигом, не сможем уйти из ромба, не посетив крайнюю левую вершину.

Допустим, первый ромб прошли зиг-загом и ушли из него, не посетив правую вершину. Тогда в эту вершину можно прийти только справа, из следующей промежуточной вершины, но нельзя этого сделать без повторного посещения. ■

3. Задача о рюкзаке

Суть задачи состоит в том, что есть много вещей разного веса и разной стоимости. Имеется рюкзак, который может выдержать определённый вес. Нужно заполнить рюкзак так, чтобы он выдержал, но при этом имела место максимальная стоимость содержимого. Это полная NP-трудная задача.

Более простая формулировка гласит, что нет стоимости, есть только веса. Вопрос в том, можно ли заполнить рюкзак, чтобы сумма весов была в точности равна вместимости рюкзака.

$$SUBSET-SUM = \{(n_1, \dots, n_k, N) : \exists\, s \subset \{0,1\}^k : \sum_{i:s_i=1} n_i = N\},$$

где s — это маска выбранных вещей.

Если все n_i и N записать в унарной записи, то это будет полиномиальная задача. Она будет решаться динамическим программированием. Используется полиномиальный алгоритм, который полиномиально зависит не от длин записи входа, а от чисел $n_1, \dots, n_k, N$. Такие алгоритмы в теории оптимизации называются **псевдополиномиальными**. Они работают, но для малых чисел.

По-настоящему полиномиальным алгоритм будет тогда, когда он будет полиномиально зависеть от длин этих записей, то есть от логарифма этих чисел. Но такое возможно, если только задача не лежит в NP.

Покажем, что

$$\text{3-SAT} \leqslant_p \text{SUBSET-SUM}.$$

Пусть l — количество переменных и m — количество скобок, тогда $n = 2l + 2m$.

Пусть есть $(p_1 \vee \bar{p_2} \vee p_3) \wedge (\bar{p_1} \vee p_2 \vee p_4) \wedge (\bar{p_2} \vee p_2 \vee p_4)$.

Покажем всё в таблице 3.1 (числа в десятичной записи). Количество разрядов в N будет равно сумме количества переменных и количества скобок.

Док-во: Пусть есть выполняющий набор. Тогда если переменная p_1 истинная, то из

	Старшие разряды (соответствуют переменным)					Младшие разряды (соответствуют скобкам)				
	p_1,	p_2,	p_3,	...,	p_l	C_1,	C_2,	C_3,	...,	C_m
n_1	1	0	0	...	0	1	0	0	...	
n_2	1	0	0	...	0	0	1	0	...	
n_3		1	0	...	0	0	1	0	...	
n_4		1	0	...	0	1	0	1	...	
n_5			1	...	0	1	0	1	...	
n_6			1	...	0	0	0	0	...	
n_7						0	1	0	...	
n_8						0	1	0	...	
			...					...		
n_{2l-1}					1					
n_{2l}					1					
n_{2l+1}						1	0	0	...	0
						1	0	0	...	0
							1	0	...	0
							1	0	...	0
								...		
$n_{2l+2m-1}$										1
n_{2l+2m}										1
N	1	1	1	...	1	3	3	3	...	3

Таблица 3.1

соответствующих двух строчек возьмём верхнюю (n_1), если ложная, то возьмём нижнюю (n_2). Если всё сложить, то внизу в разрядах N будут единицы. Так как набор выполняющий, то в каждой скобочке хотя бы один литерал истинный. Например, если p_1 — истина в C_1, то берётся верхнее число, поэтому одна единица уже есть. Если p_2 — отрицание истины, тогда p_2 — ложная, и была бы выбрана единица из четвёртой строчки. Так как в каждом дизъюнкте хотя бы один литерал истинен, то это значит, что в каждом C_i столбце будет хотя бы одна единица. Теперь, чтобы добрать до троек, имеются единицы в правом нижнем блоке. В любом случае не будет переносов, так как в первых разрядах по две единицы, а во вторых — по пять.

Пусть было набрано ровно по N-строке. Тогда из пар чисел n_1, n_2 взяли ровно одно. Если бы взяли ни одного, то был бы ноль, если оба — было бы два, а так как стоит единица, то ровно одно. Соответственно, если верхнее — переменная истинная, если нижнее — ложная.

В младших разрядах было набрано по тройке. За счёт только нижней части нельзя набрать три, максимум два, значит, в каждом C_i столбце отмечена единица в верхней части. Так как единица отмечена, то берем верхнюю строчку, и тогда истинная переменная делает истинным соответствующий дизъюнкт. И наоборот, если взяли нижнюю, то ложная переменная тоже делает дизъюнкт истинным. Значит, это действительно выполняющий набор — в каждой скобочке будет истинный литерал. ■

4. Задача поиска

С каждой задачей из NP связана задача поиска.

Пусть $V(x, s)$ — полиномиальный от длины x алгоритм. Тогда **задача поиска** заключается в нахождении s по x таким, что $V(x, s) = 1$, если такое s существует. То есть по выполнимой формуле найти выполняющий набор. Если такого s не существует, есть два варианта:

- Сложный: найти такое s, либо доказать, что его нет.
- Простой:[1] ...

[1] На этом моменте видео лекции внезапно обрывается