1 Вводная часть

Используемая терминология может отличаться от основной части работы.

Битовые строки

Булева функция отождествлена с множеством ее выполняющих наборов. Каждый выполняющий набор это битовая строка, то есть строка, состоящая из 0 и 1. **В этой работе индексация строк идет слева направо.**

Вспомогательные символы

В работе часто используется символ ? в битовой строке - это произвольный бит 0 или 1. Такую же роль могут выполнять переменные.

Сохранение общности в доказательствах

Очевидно, что порядок следования выполняющих наборов может быть любым. Более того, порядок следования переменных может быть любым, поэтому допустимы перестановки столбиков исходных битовых строк.

И даже более того, символы 0 и 1 в битовых строках можно менять местами, то есть все 0 заменять на 1, а все 1 заменять на 0 для конкретной переменной.

Фиксация переменных

В случаях 1.1, 1.2, 1.3, 1.4 игрок F должен зафиксировать пару переменных для себя, а пару отдать противнику. Но будет удобнее полагать, что в 1.2 и 1.4 он выбирает пару для противника, а другую пару берет себе.

2 Случай 1.1

Сначала F фиксирует набор подконтрольных переменных. F придает значение всем своим переменным сразу, потом тоже самое делает V.

Фиксация пары переменных первым ходом равносильна разделению всего множества наборов на 4 группы. Присвоение значения этой паре переменных равносильно выбору конкретной группы из этого разделения.

Покажем, что если выполняющих наборов ровно 4, то F имеет стратегию для победы.

Действительно, пусть зафиксированная пара переменных произвольная, НУО положим, что это x_1, x_2. Все выполняющие наборы разделятся по 4 сформированным группам, F выиграет, если хотя бы в одной из групп не будет выполняющих наборов, потому что первым ходом он выберет эту группу.

Таким образом, единственный случай, которой не подходит для F - в каждой группе находится ровно 1 выполняющий набор. Далее покажем, как перевыбрать фиксируемые переменные так, чтобы в одной из групп получилось 0 выполняющих наборов или, что равносильно, в какой-то из групп не в точности один набор.

Пусть в каждой из групп один выполняющий набор. Тогда без учета порядка эти наборы должны иметь вид:

00??

01??

10??

11??

Теперь пусть F пытается зафиксировать x_3 вместо x_2, возможны различные случаи, ключевое значение имеет количество единиц и нулей в выполняющих наборах при x_3, а также их положение:

Первый: количество единиц не равно количеству нулей, тогда какая-то пара значений повторяется и для фиксации можно выбрать переменные x_2 и x_3 (а можно и x_1 и x_3), тогда в соответствующей группе будут два набора, а следовательно в какой-то группе их будет ноль, пример:

001?
011?
101?
110?

Второй: количество нулей равно количеству единиц, но все нули в x_3 стоят рядом с нулями в x_1 (или только с единицами, два случая ниже), тогда для фиксации надо выбрать переменные x_1, x_3, в двух группах окажется по два набора, а в двух ноль:

000?	001?
010?	011?
101?	100?
111?	110?

Третий: количество нулей равно количеству единиц, но оказывается, что значения при x_3 равны таковым при x_2 во всех наборах, тогда для фиксации надо выбрать переменные x_2, x_3:

000?
011?
100?
111?

Четвертый: количество нулей равно количеству единиц и значения в x_3 противоположны x_2, тогда для фиксации надо выбрать переменные x_2, x_3:

001?
010?
101?
110?

Пятый: оставшиеся два варианта:

001?	000?
010?	011?
100?	101?
111?	110?

Они являются более интересными. Можно заметить, что при фиксации x_1, x_3 или x_2, x_3 все наборы попадают в разные группы. Поэтому в данном случае требуется рассмотреть все варианты еще и для x_4. Сразу исключим первые 4 тривиальных случая, получим 4 варианта:

0011	0000	0010	0001
0100	0111	0101	0110
1000	1011	1001	1010
1111	1100	1110	1101

Во всех четырех случаях фиксация последней пары приведет к тому, что одна из групп будет пустой.

3 Случай 1.2

Сначала F фиксирует набор подконтрольных переменных. V придает значение всем своим переменным сразу, потом тоже самое делает F.

F все еще разделяет множество всех наборов на 4 группы по 4 набора, а задача V - выбрать набор, который состоит только из единиц. Для простоты будем считать, что фиксируемые переменные F задает для V, чтобы аналогии с прошлым случаем была более явной. Покажем, что при 10 выполняющих наборах у F есть стратегия.

Переформулируем доказательство на языке битовых строк, каждый выполняющий набор будем считать битовой строкой. Пусть даны 10 битовых строк размера 4, требуется найти пару индексов, такую, что количество строк в исходном наборе с каждым фиксированным значением битов на этих индексах (перебор всех пар 00, 01, 10, 11) меньше или равно трем.

Пример

Последнее утверждение может прозвучать запутанным, поэтому подкрепим его примером, рассмотрим выполняющие наборы некоторой булевой функции как набор битовых строк:

$$0000, 0001, 0010, 0100, 1000$$
$$1111, 1110, 1101, 1011, 0111$$

Предположим, что мы бы заявили, что первые два индекса - искомые, тогда нужно подсчитать сколько строк в наборе соответствуют конкретным их значениям:

Вид строки	00??	01??	10??	11??
Количество	3	2	2	3

Можно заметить, что максимальное значение в нижней строке это 3. Поэтому разделение по группам по x_1, x_2 не привело бы к созданию группы, в которую входят только выполняющие наборы, поэтому такое разделение действительно желаемое для F.

Но эту задачу решать трудоемко. Рассмотрим двойственную к ней, то есть будем рассуждать о не выполняющих наборах или о дополнении к множеству битовых строк. В этом множестве будет всего 6 строк.

Сформулируем двойственное утверждение. Даны ровно 6 битовых строк длинны 4, нужно найти пару индексов такую, что существует минимум одна строка для каждого фиксированного значения пары.

Пример

Приведем пример, набор строк, двойственный к примеру, который был показан ранее:

$$0011, 0101, 0110, 1010, 1100, 1001$$

Составим аналогичную таблицу:

Вид строки	00??	01??	10??	11??
Количество	1	2	2	1
Прошлая таблица	3	2	2	3
Сумма	4	4	4	4

Сумма количества битовых строк с каждой фиксированной парой начальных битов в двойственном и исходных наборах всегда равна 4, потому что для каждой пары начальных битов существует ровно 4 строки, которые с них начинаются.

Теперь начнем доказывать сформулированное двойственное утверждение. Зафиксируем произвольную пару индексов (переменных), НУО индексы 1, 2 (x_1, x_2) считая слева направо. Теперь будем вычислять количество битовых строк, которые начинаются с каждого фиксированного значения пары (как в примере). Пусть эти количества отсортированы по убыванию, потому что порядок групп не имеет значения. Для F плохими можно считать всего три случая:

$$3, 2, 1, 0 \quad 2, 2, 2, 0 \quad 3, 3, 0, 0$$

Тут V всегда выбирает последнюю группу и F проигрывает. Каждый из этих случаев следует рассмотреть отдельно. Заметим, что случай, где в какой-то из групп есть все 4 возможные набора опущен, так как он является тривиальным (достаточно перевыбрать x_3, x_4).

Случай 3, 3, 0, 0

Пусть x, y, z, w это ноль или один, тогда набор битовых строк можно представить так:

$$xy??, xy??, xy??$$
$$zw??, zw??, zw??$$

Причем важно заметить, что либо $x \neq z$ либо $y \neq w$. А также вне зависимости от того, какие именно 3 битовые строки находятся в группах, в каждой группе будет минимум одна строка с 1 в третей позиции и минимум одна строка с нулем в ней же, то же самое можно сказать и про четвертый индекс.

Поэтому достаточно найти позицию в которой строки xy, zw различаются и зафиксировать её, а вторым индексом выбрать любой из пары 3,4.

Пример

Для краткости в примере сразу работаем с двойственным набором. Пусть даны 6 строк:

$$0000, 0001, 0010, 1000, 1011, 1001$$

Подсчитаем значения указанной ранее таблицы, выполнив разделение по первым двум индексам:

Вид строки	00??	01??	10??	11??
Количество	3	0	3	0

Упорядочим и получим набор 3, 3, 0, 0, что подходит под рассматриваемый случай.
Находим среди первых двух индексов различающуюся пару (сравнивая 10 и 00), имеем различие в первой позиции. Также выберем любой индекс из пары 3, 4, пусть выберем 3. Итого нужно отдать противнику переменные x_1, x_3 Проверим правильность после смены индексов:

Вид строки	0?0?	0?1?	1?0?	1?1?
Количество	2	1	2	1

Что и ожидали увидеть, нулевых групп не осталось, поэтому V не сможет сделать удачный ход. (Пример проведения полной игры будет в конце параграфа)

Случай 2, 2, 2, 0

Данный случай похож логически на предыдущий, пусть $x_1, x_2, x_3, y_1, y_2, y_3$ - ноль или единица, тогда строки будут иметь вид:

$$x_1y_1??, x_1y_1??, x_2y_2??, x_2y_2??, x_3y_3??, x_3y_3??$$

Достаточно для начала выбрать любую пару для сравнения первых двух знаков, пусть сравним x_1y_1 и x_2y_2, пусть $x_1 \neq x_2$, тогда выбираем первый индекс для фиксации, иначе второй.

Но теперь уже не получится взять произвольный индекс из 3,4, нужно подобрать тот, на котором значения различные. Возможно сделать это не получится, но тогда следует перейти к рассмотрению другой пары первых знаков x_1y_1 и x_3y_3, затем x_2y_2 и x_3y_3.

Как минимум одна из трех пар окажется подходящей, потому что любая пара отличается либо в третьем либо в четвертом индексе. Рассмотрим граничный случай:

Пример

Пусть даны строки битовые строки:

$$0001, 0000, 0110, 0100, 1110, 1100$$

Построим таблицу:

Вид строки	00??	01??	10??	11??
Количество	2	2	0	2

С учетом порядка 2, 2, 2, 0, следовательно это рассматриваемый случай. Теперь начнем перебор.
Сопоставим 00?? и 01??. Наборы, которые начинаются на 00 ($00-01$, $00-00$) отличаются в последнем бите, а наборы на 01 ($01-10$, $01-00$) в предпоследнем, значит это не подходящая пара.
Сопоставим 00?? и 11??. Наборы, которые начинаются на 00 ($00-01$, $00-00$) отличаются в последнем бите, а наборы на 11 ($11-10$, $11-00$) в предпоследнем, значит это не подходящая пара.
Сопоставим 01?? и 11??. Тут отличия в предпоследнем бите у обеих пар, поэтому выбирать следует индексы 1 (отличие между 01 и 11) и 3 (между 10 и 00).
Проверим наши рассуждения:

Вид строки	0?0?	0?1?	1?0?	1?1?
Количество	3	1	1	1

Пустых групп нет. V не сможет сделать удачного хода при фиксации x_1, x_3.

Случай 3, 2, 1, 0

В этом примере следует отказаться от рассмотрения группы, в которую попала лишь одна битовая строка, тогда этот случай сводится к рассуждениям аналогичным $3, 3, 0, 0$. Следует лишь учесть тот факт, что выбор между 3 и 4 индексами больше не произвольный. Следует выбрать индекс, значение на котором меняется в группе из двух битовых строк, так как в группе из трех строк оно меняется на обеих позициях.

Пример

Рассмотрим набор битовых строк:

$$0000, 0001, 0010, 1000, 1001, 1111$$

Построим таблицу:

Вид строки	00??	01??	10??	11??
Количество	3	0	2	1

Таким образом, получаем рассматриваемый нами случай. Как и говорилось ранее, группу, в которую попала всего одна битовая строка можно игнорировать. 00 и 10 различаются в первом символе, поэтому первый индекс для фиксации это единица. Битовые строки из группы, в которой два элемента $(1000, 1001)$ отличаются в четвертой позиции. Итого фиксируем первый и последний индексы.
Таблица для проверки:

Вид строки	0??0	0??1	1??0	1??1
Количество	2	1	1	2

Пустых групп нет, следовательно V проиграет.

Полный пример решения

Рассмотрим полный пример игры при данной последовательности ходов. Пусть дана булева функция с 10 выполняющими наборами, перечислим их:

$$\begin{matrix} 0010 & 0011 \\ 0101 & 0111 \\ 1000 & 1001 \\ 1010 & 1011 \\ 1101 & 1111 \end{matrix}$$

Теперь перейдем к двойственным наборам для того, чтобы найти переменные, которые F должен зафиксировать на первом ходу:

$$\begin{matrix} 0000 & 0001 \\ 0100 & 0110 \\ 1100 & 1110 \end{matrix}$$

Этот же набор бы показан в примере к случаю 2,2,2,0, там мы установили, что следует фиксировать переменные x_1 и x_3 (Напоминаю, что в данном случае фиксация означает передачу этих переменных противнику, так как первым ходит он). Сделаем это и отобразим немного перестроенную таблицу истинности:

x_1	x_3	x_2	x_4	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

В этой таблице мы можем видеть, что V выберет одну из четырех под таблиц размера 4, но в каждой из них содержится 0, поэтому F всегда сможет победить.

4 Случай 1.3

Сначала F фиксирует набор подконтрольных переменных. F и V придают значения своим переменным по очереди, первым ходит F.

Переходим к рассмотрению последовательных ходов V и F, при условии что F фиксирует две переменные в начале игры. Будем доказывать, что при 5 выполняющих наборах у F есть стратегия для победы.

Заметим, что прошлый подход с группировками тут не уместен. Вместо него будем использовать такую идею: любая переменная фактически делит всю таблицу истинности на две части - часть, где она равна 1 и часть, где она равна нулю. Аналогично делятся и выполняющие наборы, если их всего 5, то бывает три случая:

0???	0???	1???
0???	1???	1???
1???	1???	1???
1???	1???	1???
1???	1???	1???

С точностью до перестановки нулей и единиц. Каждая переменная делит множество выполняющих наборов в одном из трех отношений: 2 к 3, 1 к 4, 0 к 5.

Заметим сразу, что если в наборе есть хотя бы одна переменная, которая делит выполняющие наборы в отношении 0 к 5, то F может ее зафиксировать и сразу исключить все выполняющие наборы. Аналогично можно поступить и если в наборе есть переменная, которая делит наборы в отношении 1 к 4, потому что за свой второй ход F точно сможет исключить один набор (теорема из основной части работы).

Пример

Пусть выполняющие наборы функции это

$$0000, 0001, 00111, 0110, 1100$$

Можно заметить, что первая переменная делит множество выполняющих наборов в отношении 1 к 4, поэтому F фиксирует ее и произвольную, например x_2. Первый ход F это $x_1 = 1$, в множестве выполняющих наборов остается всего один набор:

$$1100$$

Не важно, как ходит V, своим вторым ходом F поставит $x_2 = 0$ и исключит все выполняющие наборы из игры, а следовательно победит.

Следовательно, мы рассматриваем только случай, когда все переменные делят наборы в отношении 2 к 3. Тогда первым ходом F оставит всего два выполняющих набора при фиксации любой переменной, дальше возможно два случая.

Первый, эти два набора имеют еще хотя бы одно одинаковое значение при какой либо переменной, тогда F фиксирует и эту переменную и исключает оба набора своим вторым ходом.

Пример

Пусть функции имеет такие выполняющие наборы:

$$0000, 0011, 1000, 1111, 1100$$

Пусть F спланировал зафиксировать x_1 и придать ей значение 0, тогда в игре останутся только наборы:

$$0000, 0011$$

Видим, что у них есть общее значение в x_2, поэтому его тоже следует зафиксировать, чтобы потом приравнять к 1.
Итого, F зафиксирует x_1 и x_2, первый ход это $x_1 = 0$, не важно как ходит V, затем F ставит $x_2 = 1$ и исключает все наборы, а следовательно побеждает.

Второй, оставшиеся два набора имеют различное значение в каждой позиции, это значит что V не сможет своим ходом сохранить их оба, а поэтому F снова получит на своем ходу всего один выполняющий набор, который он всегда сможет исключить.

Пример

Пусть функция имеет такие выполняющие наборы:

$$0000, 0111, 1000, 1111, 1100$$

Пусть F планирует фиксировать x_1, и присвоить ей значение 0, тогда в игре останутся наборы:

$$0000, 0111$$

Которые различаются во всех позициях, поэтому неважно какую позицию фиксировать второй, пусть x_4.
Итого, F фиксирует x_1, x_4, первый ход это $x_1 = 0$, V сам исключит одну из своих строк вторым ходом, пусть его ход это $x_2 = 1$, тогда в игре только один набор - 0111, который можно исключить ходом $x_4 = 0$, тогда V проиграет.

5 Случай 1.4

Сначала F фиксирует набор подконтрольных переменных. F и V придают значения своим переменным по очереди, первым ходит V.

Докажем, что у F есть стратегия для победы, если количество выполняющих равно 9, при этом поступим аналогично случаю 1.2 и перейдем к 7 двойственным наборам, наша задача - сохранить один из них.

Идея заключается, в том, чтобы отдать V пару переменных, которая принимает все возможные значения $(00, 01, 10, 11)$. Однако важным дополнительным условием является то, что в свой ход F не должен испортить себе игру, рассмотрим такой пример.

Пример

Пусть дан (уже двойственный) набор битовых строк:

$$0000, 0001, 0010, 0011, 0100, 1000, 1111$$

Можем по аналогии с 1.2 построить таблицу при фиксации первых двух переменных:

Вид строки	00??	01??	10??	11??
Количество	4	1	1	1

Вроде бы для каждой комбинации начальной пары битов нашелся свой набор, так что можно отдать x_1, x_2.
Но это ошибочное рассуждение, на первом ходу V может играть так $x_1 = 1$, тогда в игре остаются наборы:

$$1000, 1111$$

Проблема заключается в том, что любой ход F приведет к исключению одного из наборов, поэтому V сможет победить на этом наборе.
НО это говорит лишь о несостоятельности приведенной стратегии игры за F, ведь решение отдать первые две переменные кажется неправильным - V резко сокращает количество двойственных наборов.

Предлагается разделить рассмотрение этого случая на три подслучая, в первом мы сможем воспользоваться привычной стратегией, а вот другие два будут требовать иного подхода. С помощью алгоритма, описанного в рамках случая 1.2 выберем пару переменных, каждому значению которой соответствует хотя бы одна строка из набора (в данном случае строк 7, а в 1.2 было 6, поэтому одну из них можно просто игнорировать), НУО пусть это x_1, x_2.

Построим немного иную таблицу, более информативную для последовательной игры, ее общий вид:

-	0	1
0	Кол-во строк вида 00??	Кол-во строк вида 01??
1	Кол-во строк вида 10??	Кол-во строк вида 11??

Приведем пример её построения.

Пример

Рассмотрим тот же набор строк:

$$0000, 0001, 0010, 0011, 0100, 1000, 1111$$

Предложенная таблица будет иметь вид:

-	0	1
0	4	1
1	1	1

После первого хода V в игре останется одна из строк или один из столбиков этой таблицы.

Такая конструкция нужна, потому что с ее помощью удобнее рассматривать различные случаи, далее это будет видно. Первый случай, например, звучит так - *нет столбика или строки, где стоят две единицы*, его мы и рассмотрим.

Нет столбика или строки, где две единицы

Это значит, что после первого хода V в игре все еще есть хотя бы три набора, причем одна из позиций (первая или вторая) у всех наборов уже одинаковая. Задача F - присвоить значение x_3 или x_4, так чтобы в позиции, зафиксированной за V оставшиеся строки имели различное значение, тогда за свой ход он не сможет исключить все строки разом, а поэтому победит F, покажем, что это возможно всегда. Пусть x, y, z это ноль или один, НУО на первом ходу V придал значение для x_1, тогда наборы имеют такой вид:

$$xy??, xy??, xz??$$

Тут $y \neq z$. Действительно, пусть одна из двух срок $xy??$ и $xz??$ различаются в третей и четвертой позициях, иначе следует придать значение позиции, в которой они совпали. Тогда $xz??$ не может различается с оставшейся строкой во всех позициях, потому что тогда две строки вида $xy??$ полностью бы совпали (у каждой строки только одна обратная).

Это значит, что после своего хода F всегда сможет оставить пару строк вида $xy??, xz??$, поэтому V не сможет исключить их разом.

Пример

Рассмотрим такие выполняющие наборы:

$$0000, 0010, 0011, 1000, 0111, 1100, 1111$$

Вообще говоря, общий алгоритм включает в себя действия, описанные в 1.2, но для краткости пока что опустим их, и заметим, что можно зафиксировать x_1, x_2. Построим таблицу:

-	0	1
0	3	1
1	1	2

Имеем рассматриваемый случай. Отдаем противнику переменные x_1 и x_2, Всего у V есть 4 варианта для хода, опишем их все:

$x_1 = 0$	$x_1 = 1$	$x_2 = 0$	$x_2 = 1$
0000	1000	0000	0111
0010	1100	0010	1100
0011	1111	0011	1111
0111		1000	
$x_3 = 1$	$x_3 = 0$	$x_3 = 0$	$x_3 = 1$
0011	1000	0000	0111
0111	1100	1000	1111

Остается только заметить, что ни в одном из этих случаев V не удается исключить оба набора после выбора F, поэтому F побеждает.

В таблице есть 4

Фактически этот случай НУО (можно переименовать аргументы функции $1 \to 0$ и $0 \to 1$), равносилен получению такой таблицы:

-	0	1
0	4	1
1	1	1

А следовательно двойственные наборы имеют вид:

$$0000$$
$$0001$$
$$0010$$
$$0011$$
$$01??$$
$$10??$$
$$11??$$

Заметим, что в таком случае противнику можно также отдать переменные x_3 и x_4. Теперь будем строить таблицу при фиксации x_3 и x_4, но из-за того, что ряд наборов неизвестен придется рассмотреть несколько случаев. Будем перебирать последнюю пару битов у последних трех наборов.

Первый. Все три пары различные. Тогда будем иметь таблицу вида

-	0	1
0	2	2
1	2	1

Единица не обязательно в правом нижнем углу, но она обязательно одна, поэтому этот случай тривиальный.

Второй. Все три пары одинаковые, таких случая всего 4, явно их переберем:

0000	0000	0000	0000
0001	0001	0001	0001
0010	0010	0010	0010
0011	0011	0011	0011
0100	0101	0110	0111
1000	1001	1010	1011
1100	1101	1110	1111

Все эти случаи являются в некотором смысле вырожденными, потому что выбиваются из общей схемы рассуждений, их придется рассматривать отдельно. Приведем пример для первого случая (самый левый в таблице), остальные можно рассмотреть аналогично.

Зафиксируем x_1, x_2 для V, оставим x_3, x_4 для F. Теперь рассмотрим все возможные ходы в игре:

$x_1 = 0$	$x_1 = 1$	$x_2 = 0$	$x_2 = 1$
0000	1000	0000	0100
0001	1100	0001	1100
0010		0010	
0011		0011	
0100		1000	
$x_3 = 0$	$x_3 = 0$	$x_3 = 1$	$x_3 = 1$

В последней строке записан ответный ход для F. Нетрудно видеть, что F действительно побеждает в этих случаях.

Третий. Две из трех строк имеют одинаковые окончания. Таблица для такого случая:

-	0	1
0	3	2
1	1	1

Тройка не обязательно находится в левом верхнем углу, но двойка обязательно рядом с тройкой (иначе случай тривиальный).

Тогда мы можем уточнить вид нашего множества двойственных наборов:

$$\begin{gathered} 0000 \\ 0001 \\ 0010 \\ 0011 \\ 01pq \\ 10pq \\ 11xy \end{gathered}$$

Важно, что наборы pq и xy должны совпадать хотя бы в одной позиции, иначе случай оказался бы тривиальным при фиксации x_3, x_4.

Пусть мы отдали противнику x_1, x_2, тогда V не выгодно делать ход вида $x_j = 0$, потому что он оставит в игре пять наборов, у которых одна из позиций совпадает, среди таких F сможет сохранить пару с различными значениями в позиции, контролируемой V. Действительно, ведь первые 4 строки остаются игре, у них есть все возможные пары последних битов, поэтому F всегда найдет нужную.

Значит V всегда сделает хода вида $x_j = 1$, таким ходом он оставит в игре либо два набора с одинаковой парой последних битов (pq), тогда любой ход F очевидно оставит в игре оба набора. Либо же оставит наборы, которые отличаются в одной позиции, но тогда F все равно сможет сыграть в позиции, в которой они совпадают.

Пример

Пусть дано множество двойственных наборов:

$$\begin{gathered}1100\\1101\\1110\\1111\\0101\\1001\\0011\end{gathered}$$

Построим таблицу при фиксации x_1, x_2:

-	0	1
0	1	1
1	1	4

Имеем рассматриваемый случай. Только 4 находится на другом месте, поэтому алгоритм, приведенный выше, следует использовать в подстановке $1 \to 0$ и $0 \to 1$, тогда 4 будет в левом верхнем углу. Фиксируем x_3, x_4, перестраиваем таблицу:

-	0	1
0	1	3
1	1	2

Имеем рассматриваемый подслучай. Значит нужно F все же удается победить при фиксации x_1, x_2. Для начала рассмотрим неудачные ходы V вида $x_j = 1$:

$x_1 = 1$	$x_2 = 1$
1100	1100
1101	1101
1110	1110
1111	1111
1001	0101
$x_3 = 0$	$x_3 = 0$
1100	1100
1101	1101
1001	0101

Теперь перейдем к анализу ходов вида $x_j = 0$:

$x_1 = 0$	$x_2 = 0$
0101	1001
0011	0011
$x_4 = 1$	$x_4 = 1$
1101	1101
1001	0101

Таком образом, F всегда побеждает.

В таблице нет 4

Фактически этот случай НУО (в целом можно переименовать аргументы функции $1 \to 0$ и $0 \to 1$), равносилен получению такой таблице:

-	0	1
0	3	2
1	1	1

Общий вид таких наборов:

00??
00??
00??
01??
01??
10??
11??

Найдем позицию, в которой различаются наборы номер 4 и 5, пусть эта позиция - j. Рассмотрим два случая.

Первый. Последняя пара наборов различается в позиции j, тогда при фиксации x_2, x_j получим тривиальный случай.

Пример

Пусть наборы имеют вид:

0000
0011
0001
0110
0111
1000
1111

Построим таблицу:

-	0	1
0	3	2
1	1	1

В данном случае $j = 4$, так как $0110, 0111$ отличаются в четвертой позиции. В этой же позиции отличаются $1000, 1111$. Поэтому отдаем противнику x_2, x_4. Проверим таблицей при их фиксации:

-	0	1
0	2	2
1	1	2

Видим тривиальный случай

Второй. Последняя пара наборов совпадает в позиции j и равны w, Тогда отдать следует переменные x_1, x_2.

Все ходы, кроме $x_1 = 1$, оставляют в игре минимум три набора, в таком случае F всегда сможет сохранить пару такую, что в x_2 она имеет различные позиции.

Пусть ход противника $x_1 = 1$, тогда на него можно ответить $x_j = w$, сохранив оба набора.

Пример

Пусть наборы имеют вид:

$$\begin{gathered}0000\\0011\\0001\\0110\\0111\\1000\\1110\end{gathered}$$

Построим таблицу:

-	0	1
0	3	2
1	1	1

В данном случае $j = 4$, так как $0110, 0111$ отличаются в четвертой позиции. В этой позиции совпадают $1000, 1110$. Поэтому отдаем противнику x_1, x_2. Все варианты игры:

$x_1 = 0$	$x_1 = 1$	$x_2 = 0$	$x_2 = 1$
0000	1000	0000	0110
0011	1110	0011	0111
0001		0001	1110
0110		1000	
0111			
$x_3 = 1$	$x_4 = x_j = 0$	$x_3 = 0$	$x_3 = 1$

Видим, что F всегда побеждает.

6 Случай 2.1

F и V не фиксируют переменные до начала игры, а каждый раз выбирают из общего, полного набора. F и V ходят по очереди, начинает F.

Докажем, что F имеет стратегию для 3 выполняющих наборов. На самом деле это очевидно, ведь первым своим ходом он сможет исключить два из них, а вторым оставшийся. Это следствие теоремы доказанной в основной части работы.

7 Случай 2.2

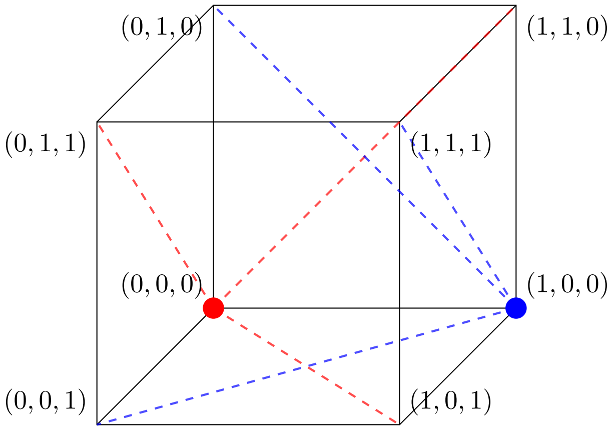
F и V не фиксируют переменные до начала игры, а каждый раз выбирают из общего, полного набора. F и V ходят по очереди, начинает V.

Докажем, что F имеет стратегию для 5 выполняющих наборов. Начнем с того, что предположим, что V уже сделал идеальный первый ход, то есть после него все еще остается 5 выполняющих наборов, но положим что их размер равен 3, потому что одна из позиций у всех одинаковая.

Перейдем к двойственным наборам, их будет всего три. Задача F - сохранить их, для этого надо чтобы после первого его хода V не смог исключить все наборы. Для этого, в свою очередь, нужно оставить либо три набора, либо два, полностью отличающихся в двух оставшихся позициях.

Предположим, что ни одна из переменных не делит множество наборов в отношении 0 к 3, потому что иначе F сохранит все три набора.

То, что мы имеем дело с наборами размера три - чудесно, потому что мы можем перейти к геометрической их интерпретации на единичном кубе. Мы хотим показать, что какие бы три точки мы не отметили на единичном кубе, найдется пара с расстоянием (Хэмминга) равным двум.



На этом доказательство фактически окончено.

Пример

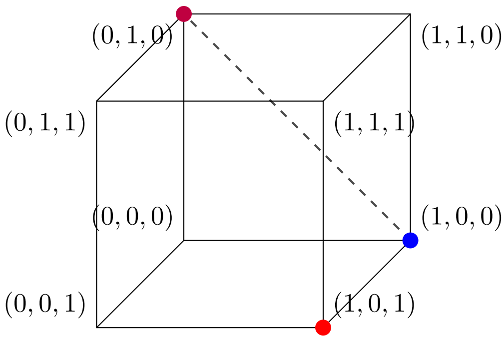
Пусть дано 5 битовых строк:

0000
0011
0001
0110
0111

Очевидно, что первый ход V это $x_1 = 0$, он сохраняет все строки. Переходим к двойственному набору:

010
100
101

На кубе:



Тогда F выбирает $x_3 = 0$ (в изначальном наборе $x_4 = 0$). Это равносильно тому, чтобы оставить только точки на одной из граней. Остаются наборы:

010
100

Видим, что $x_1 \neq x_2$, поэтому у V нет хода, который исключил бы все наборы.