Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

КУРСОВАЯ РАБОТА

на тему

Ветвящиеся программы ограниченной ширины и булевы схемы

Выполнил студент группы БПМИ145, 2 курса,
Шихова Арсения Игоревна

Научный руководитель:
Кандидат физ.-мат. наук, доцент,
Подольский Владимир Владимирович

Москва 2016

Оглавление.

Аннотация.

Цель данной работы — описать связи между классами булевых функций, вычислимых булевыми схемами или ветвящимися программами, удовлетворяющими заданным условиям. Доказываются различные оценки на вычислительные возможности классов булевых функций, вычислимых ветвящимися программами конечной ширины.

Abstract.

The purpose of this study is to describe the relationship between classes of Boolean functions computable by Boolean circuits or branching programs that meet specified conditions. We prove various bounds on classes of Boolean functions computable by branching programs of finite width.

Введение.

Булева функция от n переменных — это отображение $f : \{0,1\}^n \to \{0,1\}$, где $n \in \mathbb{N}$. Обычно булевы функции задаются в виде *формул*, состоящих из переменных, скобок и символов отрицания ($\neg$), коньюнкции ($\wedge$ — логическое «и») и дизьюнкции ($\vee$ — логическое «или»). Существуют различные модели для вычисления булевых функций, из которых наиболее стандартными являются булевы схемы и машины Тьюринга. Интересно сравнивать, насколько эффективными являются различные модели при вычислении конкретных булевых функций; давать оценки на глубину (для булевых схем), время работы (для машин Тьюринга) и так далее. Ещё одна модель для вычислений — ветвящиеся программы. Для краткого ознакомления с булевыми схемами, машинами Тьюринга и ветвящимися программами можно прочитать соответствующие разделы любого из учебников ([1], [6]).

В данной работе я изучаю связь между некоторыми классами ветвящихся программ и булевыми схемами. То есть отвечаю на вопросы вида «Если функция вычислима ветвящейся программой из такого класса, то насколько большой и сложной должна быть вычисляющая её схема?».

Ветвящиеся программы — это графы с надписями на вершинах и рёбрах, построенными по определённым правилам. Для некоторых ветвящихся программ можно определить такой параметр, как *ширина*. Цель данной работы — изучить свойства ветвящихся программ ограниченной ширины и понять, как они связаны с известными классами булевых схем.

В первой главе я определяю, что такое недетерминированные, детерминированные и перестановочные ветвящиеся программы. А также привожу доказательство теоремы Баррингтона о том, что если булева функция вычислима схемой конечной глубины, то она может быть вычислена ветвящейся программой ширины 5. Здесь возникает особый класс ветвящихся программ — *перестановочные ветвящиеся программы*.

Вторая глава состоит из двух частей. Первая – это определения. В основном из алгебры: что такое группы и моноиды, какие бывают типы групп и моноидов, какими свойствами они обладают. Далее я определяю, что такое ветвящиеся программы на моноидах — достаточно общая конструкция, включающая в себя перестановочные ветвящиеся программы, возникшие при доказательстве теоремы Баррингтона. А во второй части я рассматриваю алгоритмическую задачу — посчитать произведение элементов моноида. Посчитать выход ветвящейся программы на моноиде — частный случай этой задачи. Для различных моноидов я строю булевы схемы, позволяющие посчитать произведение элементов этого моноида.

В третьей главе я исследую два класса ветвящихся программ — перестановочные и так называемые *забывающие ветвящиеся программы*. И выясняю, как связаны перестановочные и забывающие ветвящиеся программы ширины 2, 3 и 4 с известными классами булевых схем.

Глава 1. Ветвящиеся программы. Теорема Баррингтона.

Булевы схемы и ветвящиеся программы изучаются уже давно, с начала XX века. Одни из первых исследователей в этой области — Клод Шеннон ([7]), Ли ([8]). Вообще, и булевы схемы, и ветвящиеся программы — довольно общие модели для вычисления. Мы будем изучать связи между ними для некоторых специальных случаев.

1.1. Определения.

Определение 1. *Булева функция* $f(x_1, x_2, \ldots, x_n)$ от n переменных — это отображение $f : \{0,1\}^n \to \{0,1\}$. Булева функция может не зависеть от некоторых своих переменных. Пример — функция $Maj_3(x,y,z) = 1 \Leftrightarrow x + y + z \geqslant 2$. Также под булевой функцией можно понимать любое отображение $f : \{0,1\}^n \to \{0,1\}^m$, где $n, m \in \mathbb{N}$.

Определение 2. *Булева формула* — способ записи булевой функции через *связки*: $\neg$ — отрицание, $\vee$ — конъюнкция (логическое «или») и $\wedge$ — дизъюнкция (логическое «и»). Например, $Maj_3(x,y,z) = (x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$.

Определение 3. *Булева схема* от n переменных $x_1, x_2, \ldots, x_n$ — это последовательность булевых функций $g_1, g_2, \ldots, g_t$, где $t \geqslant n$ и $\forall\, i : 1 \leqslant i \leqslant n \Leftrightarrow g_i = x_i$, а при $i > n$ g_i получается либо применением отрицания к одной из функций $g_1, g_2, \ldots, g_{i-1}$, либо применением конъюнкции или дизъюнкции к двум функциям из этого множества. Булеву схему можно рассматривать как ориентированный ациклический граф, в котором первые n вершин — *входы*, а последняя вершина — *выход*. Вершины соответствуют функциям, а рёбра — связкам. На рисунке 1.1 изображён пример булевой схемы.

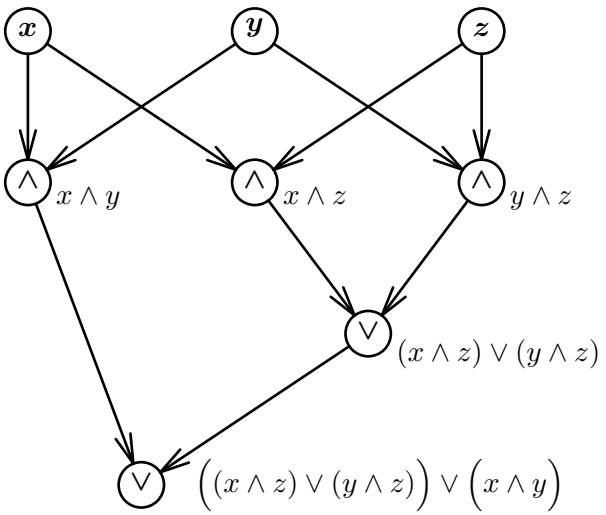

Рис. 1.1: Пример булевой схемы.

Определение 4. Булева схема C *вычисляет* булеву функцию $f : \{0,1\}^n \to \{0,1\}$, если

$\forall\ x \in \{0,1\}^n : g_t(x) = f(x)$ ($g_t(x)$ — то, что написано на выходе C).

Определение 5. *Глубина булевой схемы* — длина наибольшего пути от входа к выходу.

Определение 6. *Размер булевой формулы* — число связок в ней, *размер булевой схемы* — число её элементов (то есть t в наших обозначениях).

Определение 7. Булева формула/схема от n переменных *имеет полиномиальный размер*, если существует такой полином P, что её размер меньше или равен $P(n)$.

Определение 8. *Недетерминированная ветвящаяся программа* — это ориентированный ациклический граф. В этом графе выделены две вершины — *старт* и *выход*. У программы есть переменные $(x_1, x_2, \ldots, x_n)$. На некоторых рёбрах написаны литералы x_i (их будем называть *1-рёбрами*) или $\neg x_i$ (*0-рёбра*). Или (что то же самое) на рёбрах можно писать «$x_i = 1$» и «$x_i = 0$». Пример недетерминированной ветвящейся программы — на рисунке 1.2.

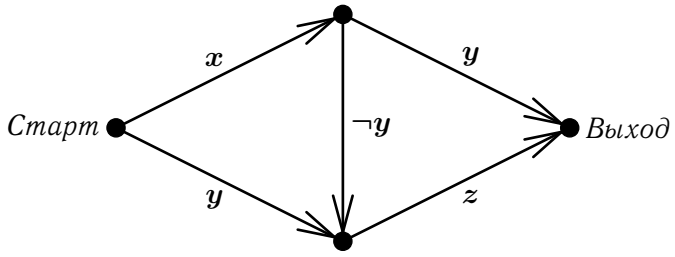

Рис. 1.2: Пример недетерминированной ветвящейся программы, вычисляющей $Maj_3(x, y, z)$ (по формуле $(x \wedge y) \vee (x \wedge \neg y \wedge z) \vee (y \wedge z)$).

Определение 9. *Недетерменированная ветвящаяся программа вычисляет булеву функцию* $f : \{0,1\}^n \to \{0,1\}$ так: пусть $a = (a_1, a_2, \ldots, a_n) \in \{0,1\}^n$. Тогда программа возвращает 1 на этом входе, если существует путь из стартовой вершины в выходную только по пустым рёбрам, рёбрам, соответствующим x_i при $a_i = 1$, и рёбрам, соответствующим $\neg x_i$ при $a_i = 0$; или 0, если такого пути не существует. На рисунке 1.2 — недетерминированная ветвящаяся программа, вычисляющая функцию $Maj_3(x, y, z)$.

Определение 10. *Детерминированная ветвящаяся программа* — это ориентированный ациклический граф. В ней выделены три вершины — *старт* и два *выхода*, 0 и 1. У программы есть переменные $(x_1, x_2, \ldots, x_n)$. Из каждой вершины, кроме выходов, выходит ровно два ребра, на которых написано $x_i = 0$ и $x_i = 1$ для некоторого i. А сама вершина помечена x_i. На рисунке 1.3 — пример детерминированной ветвящейся программы.

Определение 11. *Детерминированная ветвящаяся программа вычисляет булеву функцию* $f : \{0,1\}^n \to \{0,1\}$ так: пусть $a = (a_1, a_2, \ldots, a_n) \in \{0,1\}^n$. Тогда программа возвращает 1 на этом входе, если существует путь из стартовой вершины в выход 1 только по рёбрам, для которых $x_i = a_i$, и 0, если такого пути нету. На рисунке 1.3 — детерминированная ветвящаяся программа, вычисляющая $Maj_3(x, y, z)$.

Определение 12. *Длина* ветвящейся программы — число рёбер в самом длинном пути из стартовой вершины в выходную. Этот путь может быть *противоречивым*, то есть одновременно содержать рёбра x_i и $\neg x_i$ для некоторого i.

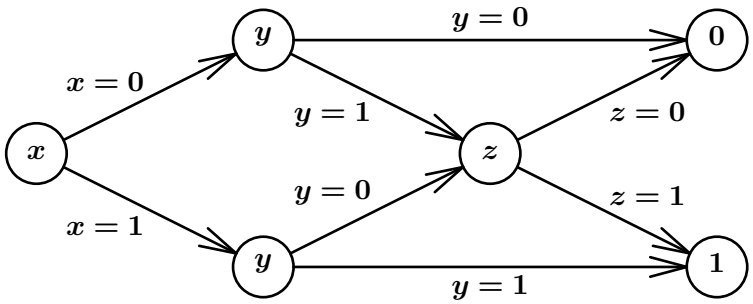

Рис. 1.3: Пример детерминированной ветвящейся программы, вычисляющей $Maj_3(x, y, z)$.

Определение 13. Ветвящаяся программа называется *уровневой*, если её вершины могут быть разделены на уровни так, что рёбра, выходящие из вершин i-го уровня, идут только в вершины $(i+1)$-го. *Размер уровня* — число вершин в нём.

Определение 14. *Ширина* уровневой ветвящейся программы — размер наибольшего уровня.

Определение 15. *Перестановка на множестве* $1, 2, \ldots, n$ — биекция этого множества в себя. Если σ — перестановка, то обозначают $\sigma = \left(\begin{smallmatrix} 1 & 2 & \cdots & n \\ \sigma_1 & \sigma_2 & \cdots & \sigma_n \end{smallmatrix}\right)$. n — *порядок перестановки*.

Определение 16. S_n — множество всех перестановок на n элементах.

Определение 17. Перестановка σ на множестве $1, 2, \ldots, n$ называется *циклом длины* l, если $\exists\ i_1, i_2, \ldots, i_l \in \{1, 2, \ldots, n\}$ такие, что $\forall\ j \notin \{i_1, i_2, \ldots, i_l\} : \sigma_j = j$, $\forall\ k, 1 \leqslant k \leqslant l-1 : \sigma_{i_k} = i_{k+1}$ и $\sigma_{i_l} = i_1$. Обозначение $\sigma = (i_1 i_2 \ldots i_l)$.

Определение 18. *Транспозиция* (или *инверсия*) — перестановка, которая меняет местами два элемента i и j. Обозначение $\tau_{ij} = \left(\begin{smallmatrix} 1 & 2 & \ldots & i & \ldots & j & \ldots & n \\ 1 & 2 & \ldots & j & \ldots & i & \ldots & n \end{smallmatrix}\right)$.

1.2. Теорема Баррингтона.

Теорема. [3] Если булева функция f может быть вычислена схемой глубины d, то она может быть вычислена уровневой ветвящейся программой ширины 5 и длины не более 4^d.

Баррингтон доказал более общий факт. Обозначим NC^1 — класс булевых функций, вычислимых схемами логарифмической глубины и полиномиального размера. Тогда NC^1 совпадает с классом булевых функций, вычислимых перестановочными ветвящимися программами конечной ширины. (О том, что такое перестановочные ветвящиеся программы, сказано в следующем разделе). А я доказываю включение в одну сторону. В самом деле, пусть булева функция f вычислима схемой глубины $c\log_2(n)$ для некоторой константы c. Тогда по теореме Баррингтона f может быть вычислена ветвящейся программой ширины 5 и длины $4^{c\log_2 n} = 4^c \cdot n^2$. Доказательство включения в другую сторону менее объёмное и трудное.

1.2.1. Перестановочные ветвящиеся программы.

Рассмотрим детерминированную уровневую программу ширины w и длины l, причём вершины разбиваются на l уровней размера ровно w каждый, и на каждом уровне все вершины соответствуют ровно одной переменной x_i. Кроме того, на каждом уровне исходящие

из него 0-рёбра и исходящие из него 1-рёбра образуют две биекции из $[w] = \{1, 2, \ldots, w\}$ в себя. То есть если какому-то уровню соответствует переменная x_i, то одно из этих отображений — рёбра, на которых написано «$x_i = 0$», а второе — рёбра, на которых написано «$x_i = 1$». Таким образом, *перестановочная ветвящаяся программа* ширины w и длины l — ветвящаяся программа, имеющая l уровней по w вершин, в которой каждому переходу между уровнями соответствуют две перестановки из S_w; любому входу $x \in \{0,1\}^n$ соответствует перестановка $P(x)$, являющаяся композицией выбираемых на каждом уровне в соответствии с входом перестановок. Значит, можем рассматривать нашу программу как множество троек $\langle i, \sigma, \tau \rangle$, где x_i — переменная, которую мы используем, а $\sigma, \tau : [w] \to [w]$ — два отображения, соответствующие 0-рёбрам и 1-рёбрам.

Определение 19. Если f — булева функция, σ — перестановка, перестановочная ветвящаяся программа P *σ-вычисляет* f, если для для любого входа x:

$$P(x) = \begin{cases} \sigma, f(x) = 1, \\ e, f(x) = 0. \end{cases}$$

Здесь e — тождественная перестановка.

Вопрос: как, имея программы, вычисляющие функции f и g, строить по ним программы, вычисляющие $\neg f, f \vee g, f \wedge g$? (этого достаточно, чтобы рекурсивно построить программу, вычисляющую любую булеву функцию). Баррингтон показал, что это можно сделать, если ужесточить условия для наших программ, используя только те, для которых все σ и τ — циклы.

1.2.2. Леммы.

Пусть P, Q — две перестановочные ветвящиеся программы от t переменных; σ, τ — циклические перестановки одинакового порядка; f, g — булевы функции от t переменных.

Лемма 1. (Поменять местами входы). Если P σ-вычисляет f, то существует перестановочная ветвящаяся программа того же размера, τ-вычисляющая f.

Доказательство.

□ Так как σ и τ — циклы, то существует такая перестановка θ, что $\tau = \theta\sigma\theta^{-1}$. Если $\sigma = \left(\begin{smallmatrix} i_1 & i_2 & \ldots & i_t \\ i_2 & i_3 & \ldots & i_1 \end{smallmatrix}\right)$, $\tau = \left(\begin{smallmatrix} j_1 & j_2 & \ldots & j_t \\ j_2 & j_3 & \ldots & j_1 \end{smallmatrix}\right)$, то $\theta = \left(\begin{smallmatrix} i_1 & i_2 & \ldots & i_t \\ j_1 & j_2 & \ldots & j_t \end{smallmatrix}\right)$.

Отсортируем вершины на первом уровне P в соответствии с перестановкой θ (каждую i-ю вершину ставим на θ_i-е место). Аналогично для последнего уровня P и перестановки θ^{-1}. Получим другую перестановочную ветвящуюся программу P'. Рассмотрим вход x. Пусть $P(x) = \sigma$, где $\sigma = \left(\begin{smallmatrix} 1 & 2 & \ldots & t \\ \sigma_1 & \sigma_2 & \ldots & \sigma_t \end{smallmatrix}\right)$. А $P'(x) = \theta\sigma\theta^{-1} = \tau$, где $\tau = \left(\begin{smallmatrix} \theta_1 & \theta_2 & \ldots & \theta_t \\ \theta(\sigma_1) & \theta(\sigma_2) & \ldots & \theta(\sigma_t) \end{smallmatrix}\right)$. Если $P(x) = e$, то $P'(x) = \theta\theta^{-1} = e$. Для любого входа x: если $P(x) = \sigma$, то $P'(x) = \tau$; а если $P(x) = e$, то $P'(x) = e$. Что и требовалось доказать. ■

Лемма 2. (Отрицание). Если P σ-вычисляет f, то существует перестановочная ветвящаяся программа того же размера, σ-вычисляющая $\neg f$.

Доказательство.

□ Так как σ — цикл, то и σ^{-1} — цикл. По предыдущей лемме существует программа P', σ^{-1}-вычисляющая f. Для входа x $P'(x) = \sigma^{-1}$, если $f(x) = 1$, и $P'(x) = e$, если $f(x) = 0$.

Переупорядочим вершины последнего уровня $P'(x)$ в соответствии с перестановкой σ. То есть каждую i-ю вершину поставим на σ_i-е место. Получим новую программу P''. Если $P''(x) = e$, то $P'(x) = \sigma^{-1}$, следовательно, $f(x) = 1$. И наоборот, если $P''(x) = \sigma$, то $P'(x) = e$, следовательно, $f(x) = 0$. $P''(x) = e \Rightarrow \neg f(x) = 0$; $P''(x) = \sigma \Rightarrow \neg f(x) = 1$. Именно такую программу и требовалось построить.

На рисунке 1.4 — пример для $f(x) = x$. ■

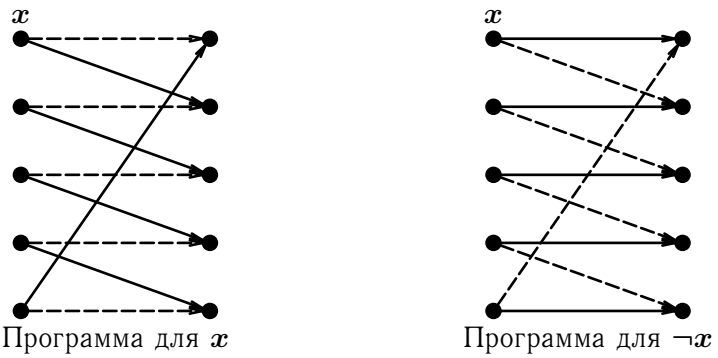

Рис. 1.4: Пример программ (ширины 5), вычисляющих x и $\neg x$ соответственно. Пунктиром показаны 0-рёбра, обычные стрелки — 1-рёбра.

Лемма 3. (Конъюнкция). Если P σ-вычисляет f и Q τ-вычисляет g (обозначим $|X|$ — количество уровней программы X), то существует перестановочная ветвящаяся программа, у которой $2(|P| + |Q|)$ уровней и которая $\sigma\tau\sigma^{-1}\tau^{-1}$-вычисляет $f \wedge g$.

Доказательство.

□ По лемме 1, существуют такие программы P' и Q', что P' σ^{-1}-вычисляет f и Q' τ^{-1}-вычисляет g. Пусть R — программа, которая последовательно выполняет P, Q, P' и Q'. Если для входа x : $f(x) = 0$ и $g(x) = 0$, то $P(x) = e, P'(x) = e, Q(x) = e$ и $Q'(x) = e$. Значит, $R(x) = e$. Если $f(x) = 0, g(x) = 1$, то $P(x) = e, P'(x) = e, Q(x) = \tau, Q'(x) = \tau^{-1}$. Значит, $R(x) = e$. Если $f(x) = 1, g(x) = 0$, то $P(x) = \sigma, P'(x) = \sigma^{-1}, Q(x) = e, Q'(x) = e$. Значит, $R(x) = e$.

Если же $f(x) = 1, g(x) = 1$, то $P(x) = \sigma, P'(x) = \sigma^{-1}, Q(x) = \tau, Q'(x) = \tau^{-1}$. Значит, $R(x) = \left(\tau^{-1}(\sigma^{-1}\overset{1}{(}\tau(\sigma_1))) \; \tau^{-1}(\sigma^{-1}\overset{2}{(}\tau(\sigma_2))) \; \overset{\cdots}{\ldots} \; \tau^{-1}(\sigma^{-1}\overset{t}{(}\tau(\sigma_t)))\right)$. Так как $|P| = |P'|, |Q| = |Q'|$, то $|R| = 2(|P| + |Q|)$, следовательно, R удовлетворяет условию.

На рисунке 1.5 — пример для $f(x, y) = x, g(x, y) = y$. ■

Лемма 4. Существуют циклические перестановки σ и $\tau \in \{1, 2, 3, 4, 5\}$ такие, что $\sigma\tau\sigma^{-1}\tau^{-1}$ тоже циклическая.

Мы будем строить по булевой функции f перестановочную ветвящуюся программу ширины 5, которая вычисляет f. Будем строить по формуле, соответствующей f — для этого надо уметь по программам, вычисляющим функции g и h, строить программы, вычисляющие $\neg g$,

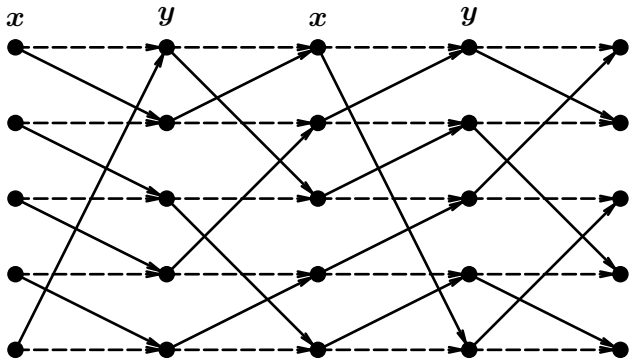

Рис. 1.5: Пример программы, вычисляющей $x \wedge y$.

$g \wedge h$, $g \vee h$. Благодаря лемме 1 можно говорить просто «программа G вычисляет функцию g» вместо «G σ-вычисляет g для некоторого σ», потому что σ может быть произвольным циклом из S_5. В лемме 3 возникает перестановка $\sigma\tau\sigma^{-1}\tau^{-1}$. Значит, нужно показать, что если σ и τ — циклы, то $\sigma\tau\sigma^{-1}\tau^{-1}$ тоже может быть циклом.

Доказательство.

□ $\sigma = (12345) = \left(\begin{smallmatrix}1&2&3&4&5\\2&3&4&5&1\end{smallmatrix}\right), \tau = (13542) = \left(\begin{smallmatrix}1&2&3&4&5\\3&1&5&2&4\end{smallmatrix}\right)$. $\sigma\tau\sigma^{-1}\tau^{-1} = (13254) = \left(\begin{smallmatrix}1&2&3&4&5\\3&5&2&1&4\end{smallmatrix}\right)$.

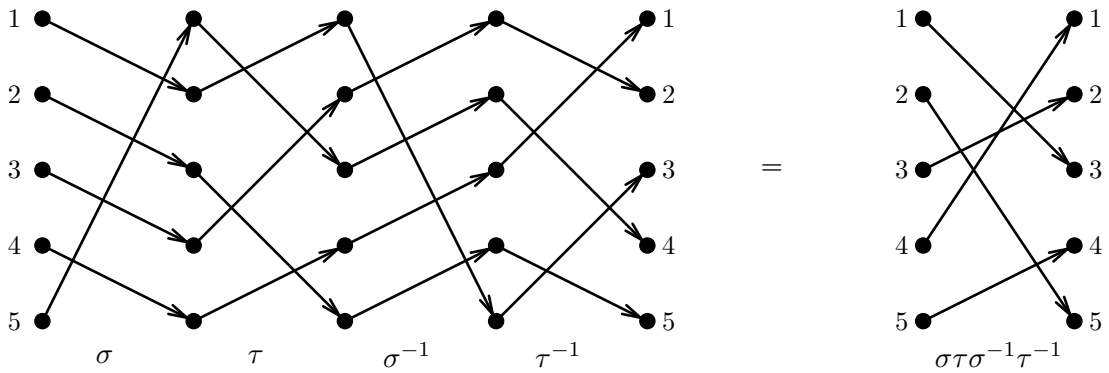

Рис. 1.6: Вместо доказательства можно нарисовать картинку.

По рисунку 1.6 видно, что $\sigma\tau\sigma^{-1}\tau^{-1}$ — тоже цикл. ■

Лемма 5. Если булева функция f может быть вычислена булевой схемой глубины d, то она может быть вычислена перестановочной ветвящейся программой ширины 5 и длины не более чем 4^d.

Доказательство.

□ Индукцией по d. Доказываем более общее утверждение: «Если булева функция f может быть вычислена булевой схемой глубины d, то существует перестановочная ветвящаяся программа P ширины 5 и длины не болеее чем 4^d такая, что P σ-вычисляет f и σ — цикл.»

База. Пусть $d = 0$, то есть выход соответствующей f схемы — один из входов. Тогда f — это либо переменная x_i, либо её отрицание $\neg x_i$. В обоих случаях соответствующая программа P имеет ширину 5 и длину $4^0 = 1$, как и требовалось (см. пример к лемме 2). При этом можно взять любую $\sigma \in S_5 \setminus \{e\}$ такую, что P σ-вычисляет f. Выберем в качестве σ цикл.

При этом можно взять любую $\sigma \in S_5 \setminus \{e\}$ такую, что P σ-вычисляет f. Выберем в качестве σ цикл.

Переход (от $d-1$ к d). Тогда верно одно из следующих утверждений:

* $f = \neg g$, причём g вычислима схемой глубины $d-1$;
* $f = g \wedge h$, причём g и h вычислимы схемами глубины $d-1$;
* $f = g \vee h$, причём g и h вычислимы схемами глубины $d-1$.

В первом случае по предположению индукции существует перестановочная ветвящаяся программа P' ширины 5 и длины не более чем 4^{d-1}, σ-вычисляющая g, где σ — цикл. Но по лемме 2 существует перестановочная ветвящаяся программа P'' ширины 5 и той же длины, σ-вычисляющая $\neg g = f$.

Во втором случае, пусть программа G σ-вычисляет g, H τ-вычисляет h. По предположению индукции, G и H имеют ширину 5 и длину максимум 4^{d-1}, σ и τ — циклы. По лемме 1, существуют программы G' и H' ширины 5 и той же длины, что G и H, такие, что G' (12345)-вычисляет g, H' (13542)-вычисляет h. Но тогда по лемме 3 существует перестановочная ветвящаяся программа F ширины 5 и длины $2 \cdot (|G'|) + |H'|) \leqslant 2 \cdot (4^{d-1} + 4^{d-1}) = 4^d$ такая, что F (13254)-вычисляет f. (13254) — цикл, значит, F подходит под условие леммы и шаг индукции доказан.

Третий случай. Заметим, что $\forall\, x, y \in \{0,1\}$: $x \vee y = \neg(\neg x \wedge \neg y)$. Таблица истинности:

x	y	$\neg x$	$\neg y$	$\neg x \wedge \neg y$	$\neg(\neg x \wedge \neg y)$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

Для $\neg g$ и $\neg h$ существуют программы длины не более чем 4^{d-1} (как и для g и h). Для $\neg g \wedge \neg h$ — длины не более чем $2 \cdot (4^{d-1} + 4^{d-1}) = 4^d$, и та же оценка для $\neg(\neg g \wedge \neg h)$. Что и требовалось доказать. ■

Теорема Баррингтона — это и есть последняя лемма.

Глава 2. Задача: посчитать произведение элементов моноида.

2.1. Определения.

2.1.1. Группы.

Определение 20. Группа G — это множество с определенной на нём операцией $\cdot$, которую условно называют умножением. Часто пишут вместо $x \cdot y$ просто xy. $\cdot$ обладает следующими свойствами:

1. $\forall\ x, y \in G : xy \in G$ — *замкнутость*;
2. $\forall\ x, y, z \in G : (xy)z = x(yz)$ — *ассоциативность*;
3. $\exists\ e \in G : \forall\ x \in G\ ex = xe = x$ — *существование нейтрального элемента*;
4. $\forall\ x \in G\ \exists\ x^{-1} \in G : xx^{-1} = x^{-1}x = e$ — *существование обратного элемента.*

Замечание. Нейтральный элемент единственен. Если бы существовали два нейтральных элемента e_1 и e_2, то было бы верно равенство $e_1 = e_1e_2 = e_2$.

Определение 21. Группа *тривиальна*, если она состоит из одного элемента (единицы).

Определение 22. *Порядок* группы G — число её элементов (обозначается $|G|$). Группа называется *конечной*, если её порядок конечен, и *бесконечной* иначе.

Определение 23. Пусть G — группа. *Порядок* элемента $g \in G$ — наименьшее целое число $m \geqslant 0$ такое, что $g^m = e$. Обозначение $m = \operatorname{ord} g$.

Определение 24. Пусть G — группа, $H \subseteq G$. H — *подгруппа* G (обозначение $H \leqslant G$), если H — группа относительно $\cdot$.

Определение 25. Пусть G — группа, $H \leqslant G$. Если $H \neq G$, то H — *собственная подгруппа* G, иначе — *несобственная*.

Определение 26. Группа G называется *циклической*, если $\exists\ g \in G$: $G = \{g^k \mid k \in \mathbb{N} \cup \{0\}\}$. Тогда G *порождена* элементом x.

Определение 27. Пусть G — группа, $H \leqslant G$, $g \in G$. *Левым смежным классом* элемента g группы G по подгруппе H называется множество $gH = \{gh \mid h \in H\}$. Аналогично определяется *правый смежный класс*.

Определение 28. Пусть G — группа, $H \leqslant G$. H — *нормальная подгруппа* G (обозначение $H \lhd G$), если $\forall\ g \in H : gH = Hg$.

Лемма 6. Пусть G — группа, $H \leqslant G$ и $g_1, g_2 \in G$. Тогда либо $g_1H = g_2H$, либо $g_1H \cap g_2H = \varnothing$.

Доказательство.

$\square$ Пусть $g_1H \cap g_2H \neq \varnothing$, то есть $\exists\ h_1, h_2 \in H : g_1h_1 = g_2h_2$. $g_1H = g_2h_2h_1^{-1}H \subseteq g_2H$, аналогично, $g_2H \subseteq g_1H$. Значит, $g_1H = g_2H$. $\blacksquare$

Лемма 7. Пусть G — группа, H — конечная подгруппа G. Тогда $\forall\ g \in G : |gH| = |H|$.

Доказательство.

□ По определению смежного класса, $gH = \{gh \mid h \in H\}$. Значит, $|gH| \leqslant |H|$. Предположим, что $\exists\ h_1, h_2 \in H : gh_1 = gh_2$. Тогда домножим обе части равенства на g^{-1} и получим $h_1 = h_2$. Значит, все элемента вида gh, где $h \in H$, попарно различны и $|gH| = |H|$. ■

Теорема. (Лагранжа) Пусть G — конечная группа, $H \leqslant G$. Тогда $|G| \vdots |H|$.

Доказательство.

□ Каждый элемент группы G лежит в своём смежном классе по подгруппе H. Разные смежные классы не пересекаются. И каждый из них содержит ровно $|H|$ элементов. ■

Лемма 8. Пусть G — конечная группа, $g \in G$. Тогда $|G| \vdots \operatorname{ord} g$.

Доказательство.

□ Обозначим $m = \operatorname{ord} g$. $\forall\ n \in \mathbb{Z}$, если $n = mq + r$ (деление с остатком), то $g^n = (g^m)^q \cdot g^r = e^q \cdot g^r = g^r$. Рассмотрим последовательность $e, g, g^2, \ldots, g^{m-1}$. Предположим, среди этих элементов G есть два равных — g^i и g^j, где $i < j$. Значит, $e = g^i \cdot (g^i)^{-1} = g^j \cdot (g^i)^{-1} = g^j \cdot g^{m-i} = g^{m+j-i} = g^{j-i}$; $0 < j - i < m$, что противоречит определению порядка. Значит, $\{e, g, g^2, \ldots, g^{m-1}\}$ образуют подгруппу $|G|$ порядка m, и по теореме Лагранжа $|G| \vdots m$. ■

Определение 29. Пусть G — группа, $H \lhd G$, G/H — множество левых смежных классов G по H. Тогда G/H называется *факторгруппой* G по H.

Лемма 9. G/H является группой с определённой на нём операцией $(g_1H)(g_2H) = ((g_1g_2)H)$, где $g_1, g_2 \in G$.

Доказательство.

□ Замкнутость следует из замкнутости $\cdot$ на G, ассоциативность — из ассоциативности $\cdot$. Нейтральным будет элемент eH, а обратным к gH — $g^{-1}H$. ■

Определение 30. Пусть G — группа. Тогда её *нормальный ряд* G — последовательность её подгрупп $G_0, G_1, \ldots, G_s$ такая, что $\{e\} = G_s \lhd \ldots \lhd G_1 \lhd G_0 = G$.

Определение 31. Пусть G — группа. Тогда её *композиционный ряд* — её нормальный ряд такой, что $\forall\ i : G_{i+1}$ — наибольшая несобственная нормальная подгруппа G_i. Соответствующие факторгруппы G_{i+1}/G_i — *композиционные факторы* ряда.

Лемма 10. Если G — конечная группа, то у G есть композиционный ряд.

Доказательство.

□ Пусть H — произвольная подгруппа G. $\{e\}$ — нормальная подгруппа H, так как $\forall\ x \in H : \exists\ y = x \in H : ex = ye$. Значит, у любой подгруппы G, кроме $\{e\}$, есть хотя бы одна несобственная подгруппа (т. е. $\{e\}$), а значит, есть и наибольшая. Каждый раз при построении очередного элемента композиционного ряда число элементов группы становится строго меньше. Так как изначально оно конечно, то рано или поздно дойдем до $\{e\}$. ■

Определение 32. Пусть G — группа, $g, h \in G$. Тогда их *коммутатор* — $g^{-1}h^{-1}gh$, обозначается $[g, h]$.

Определение 33. Пусть G — группа. Тогда её коммутатор $G' = \{[g, h] \mid g, h \in G\}$ — подгруппа G, образованная коммутаторами всех её элементов.

Замечание. По определению, коммутатор группы — это просто некоторое множество с определённой на нём операцией. То, что оно само является группой, надо доказывать.

Лемма 11. Пусть G — группа, H — её коммутатор. Тогда $H \lhd G$.

Доказательство.

□ См. доказательство теоремы 2.23 книги [5]. ■

Определение 34. Пусть G — группа. Обозначим $G^{(0)} = G$, $\forall\ i > 0 : G^{(i)} = (G^{(i-1)})'$. G *разрешимая*, если $\exists\ i \geqslant 0 : G^{(i)} = \{e\}$.

Лемма 12. Пусть группа G конечна и разрешима. Тогда члены её композиционного ряда — циклические группы порядков, являющихся простыми числами.

Доказательство.

□ Следует из теорем, доказанных в этой книге [5]. ■

Определение 35. Пусть G, H — группы. G и H *изоморфны*, если $\exists\ \varphi : G \to H$ — такое отображение, что $\forall g_1, g_2 \in G : \varphi(g_1 g_2) = \varphi(g_1)\varphi(g_2)$ и φ — биекция. Обозначение $G \cong H$.

2.1.2. Моноиды.

Определение 36. Моноид M — это множество с определенной на нём операцией $\cdot$, которую условно называют умножением. Часто пишут вместо $x \cdot y$ просто xy. $\cdot$ обладает следующими свойствами:

1. $\forall\ x, y \in M : xy \in M$ — *замкнутость*;
2. $\forall\ x, y, z \in M : (xy)z = x(yz)$ — *ассоциативность*;
3. $\exists\ e \in M : \forall\ x \in M\ ex = xe = x$ — *существование нейтрального элемента*. Часто e называют *единицей* и обозначают 1.

Замечание. Как и в группе, нейтральный элемент в моноиде единственен.

Определение 37. Пусть M — моноид, $N \subseteq M$. Тогда N — *подмоноид* M (обозначение $N \leqslant M$), если N является моноидом относительно $\cdot$.

Определение 38. Пусть M — моноид, $N \leqslant M$ и N — группа относительно $\cdot$. Тогда N — *подгруппа* M.

Определение 39. Пусть M — моноид, $N \leqslant M$. N — *собственный подмоноид* M, если $N \neq M$.

Определение 40. Моноид называется *непериодическим*, если у него нет никаких подгрупп, кроме тривиальной.

Определение 41. Моноид называется *разрешимым*, если все его подгруппы разрешимы, и *неразрешимым* иначе.

Определение 42. Моноид M называется *циклическим*, если $\exists\ x \in M$: $M = \{x^k \mid k \in \mathbb{N} \cup \{0\}\}$.

Определение 43. Пусть M — моноид, $J \subseteq M$. J — *левый идеал* M, если $MJ \subseteq J$. Если же $JM \subseteq J$, то J — *правый идеал*.

Определение 44. Моноид M называется *моноидом левых нулей*, если $\forall x, y \in M, x \neq 1 :$ $xy = x$. Аналогично определяется *моноид правых нулей*.

2.1.3. Булевы схемы, ветвящиеся программы на моноидах.

Определение 45. $MAJ_n(x) = 1 \Leftrightarrow x_1 + \ldots + x_n > \lfloor n \,/\, 2 \rfloor$, где $x = (x_1, \ldots, x_n) \in \{0,1\}^n$. $MOD_m(x) = 1 \Leftrightarrow x_1 + \ldots + x_n \equiv 0 \pmod m$, где $x = (x_1, \ldots, x_n) \in \{0,1\}^n$.

Замечание. У нас уже было определение булевой схемы, где мы считали, что все элементы, кроме входных, получены отрицанием, конъюнкцией или дизъюнкцией каких-то предыдущих элементов. Это на самом деле частный случай булевой схемы.

Определение 46. *Булева схема* с n входами и m выходами — это связный ациклический ориентированный мультиграф $G = (V, A)$, в котором выделены m упорядоченных вершин-выходов. Вершины-входы — те, у которых входная степень равна 0. l — функция, ставящая в соответствие каждой вершине метку на ней. Если для вершины g $v_{in} = 0$, то $l(g) \in \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n, 0, 1\}$. Если же $v_{in} = k \geqslant 1$, то $l(g) = f_g : \{0,1\}^k \to \{0,1\}$ — булева функция.

Определение 47. Булева схема C с n входами и m выходами *вычисляет* булеву функцию $f : \{0,1\}^n \to \{0,1\}^m$ так. Пусть *вход* $a = (a_1, \ldots, a_n) \in \{0.1\}^n$. Тогда $\forall\ i$ вместо переменной x_i подставляем a_i во все формулы, написанные на вершинах. Тогда *результат вычислений* ($\in \{0,1\}^m$) — то, что написано на выходах C. На рисунке 2.1 — булева схема, вычисляющая Maj_4.

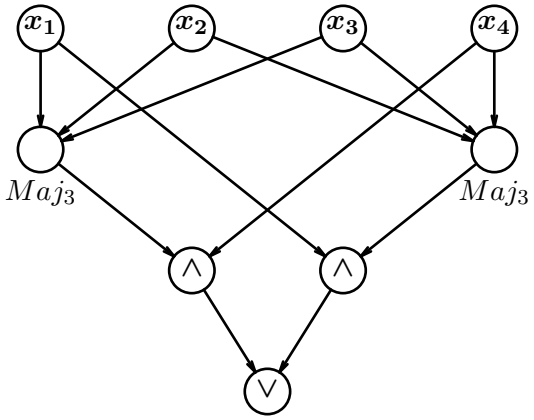

Рис. 2.1: Пример булевой схемы, вычисляющей Maj_4. Используемые связки: $\vee$, $\wedge$ и Maj_3.

Определение 48. *Размер* булевой схемы — количество её вершин.

Определение 49. *Глубина* булевой схемы — длина наибольшего пути из входной вершины в выходную.

Определение 50. AC^0 — класс булевых функций, вычислимых схемами конечной глубины и полиномиального размера, построенными только с помощью связок $\vee$ и $\wedge$ неограниченной входной степени. То есть можем посчитать конъюнкцию (дизъюнкцию) произвольного числа переменных одной связкой.

Определение 51. $AC^0[m]$ — класс булевых функций, вычислимых схемами конечной глубины и полиномиального размера, построенными только с помощью связок $\vee$, $\wedge$ (неограниченной входной степени) и MOD_m.

Определение 52. $ACC^0 = \bigcup_{m=2}^{+\infty} AC^0[m]$.

Определение 53. NC^1 — класс булевых функций, вычислимых схемами логарифмической глубины и полиномиального размера, построенных только с помощью связок $\vee$ и $\wedge$ входной степени 2.

Лемма 13. Пусть C — схема, использующая только связки $\vee$, $\wedge$ и $\neg$. Тогда можно построить схему C' такую, что $\forall\, x = (x_1, \ldots, x_n)$ на входе x C и C' дают одинаковый результат. При этом C' использует только связки $\vee$ и $\wedge$, имеет такую же глубину, что и C, и размер не более чем удвоенный размер C.

Доказательство.

□ Будем одновременно с каждой функцией, соответствующей элементу схемы, вычислять её отрицание. Для $x_1, \ldots, x_n$ и $\neg x_1, \ldots, \neg x_n$ отрицания уже содержатся во входе. Для элементов схемы, полученных конъюнкцией или дизъюнкцией, используем законы Де Моргана ($\neg(x \vee y) = \neg x \wedge \neg y$ и $\neg(x \wedge y) = \neg x \vee \neg y$, где x и y — логические переменные). На рисунках ниже — как строить элементы C', соответствующие элементам C и их отрицаниям, в зависимости от связки, с помощью которой они получены.

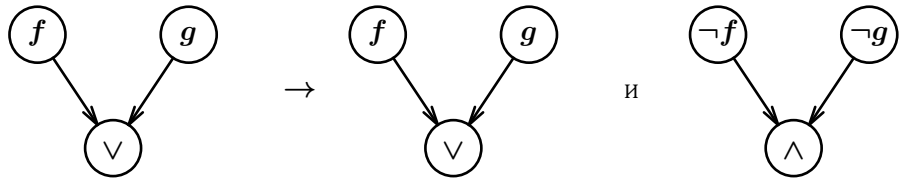

Рис. 2.2: Для дизъюнкции

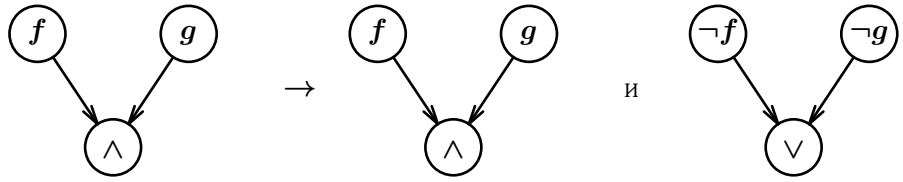

Рис. 2.3: Для конъюнкции

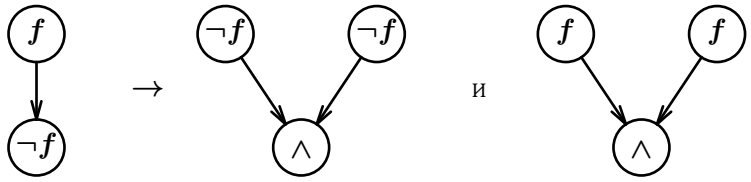

Рис. 2.4: Для отрицания

Если элемент получен связкой $\neg$, то и его отрицание, и он сам уже построены. На рисунке 2.4 — как это формально выразить. ■

Замечание. Здесь мы предполагали, что $\vee$ и $\wedge$ имеют входную степень 2. Очевидным образом можно распространить доказательство на случай неограниченной входной степени, так как $\neg(f_1 \vee f_2 \vee \ldots \vee f_n) = (\neg f_1 \wedge \neg f_2 \wedge \ldots \wedge \neg f_n)$; $\neg(f_1 \wedge f_2 \wedge \ldots \wedge f_n) = (\neg f_1 \vee \neg f_2 \vee \ldots \vee \neg f_n)$.

Лемма 14. Для любой функции $f : \{0,1\}^n \to \{0,1\}$ $\exists\ c \in N : f$ можно вычислить схемой размера не больше чем $cn2^n$, использующей только связки $\vee$ и $\wedge$.

Доказательство.

$\square$ Пусть $f : \{0,1\}^n \to \{0,1\}$, $B \subseteq \{0,1\}^n$ — такое множество, что $f(x) = 1 \Leftrightarrow x \in B$; $B = \{b_1 = b_{1.1}b_{1.2}\dots b_{1.n}, b_2 = b_{2.1}b_{2.2}\dots b_{2.n}, \dots, b_l = b_{l.1}b_{l.2}\dots b_{l.n}\}$.

Пусть $x = (x_1, x_2, \dots, x_n)$. Рассмотрим формулу $\varphi(x) = \Big((x_1 = b_{1.1}) \wedge (x_2 = b_{1.2}) \wedge \dots \wedge (x_n = b_{1.n})\Big) \vee \Big((x_1 = b_{2.1}) \wedge (x_2 = b_{2.2}) \wedge \dots \wedge (x_n = b_{2.n})\Big) \vee \dots \vee \Big((x_1 = b_{l.1}) \wedge (x_2 = b_{l.2}) \wedge \dots \wedge (x_n = b_{l.n})\Big)$. Такую формулу несложно вычислить схемой, и $\varphi(x_1, x_2, \dots, x_n) \equiv f(x_1, x_2, \dots, x_n)$. Пример — на рисунке 2.5. $\blacksquare$

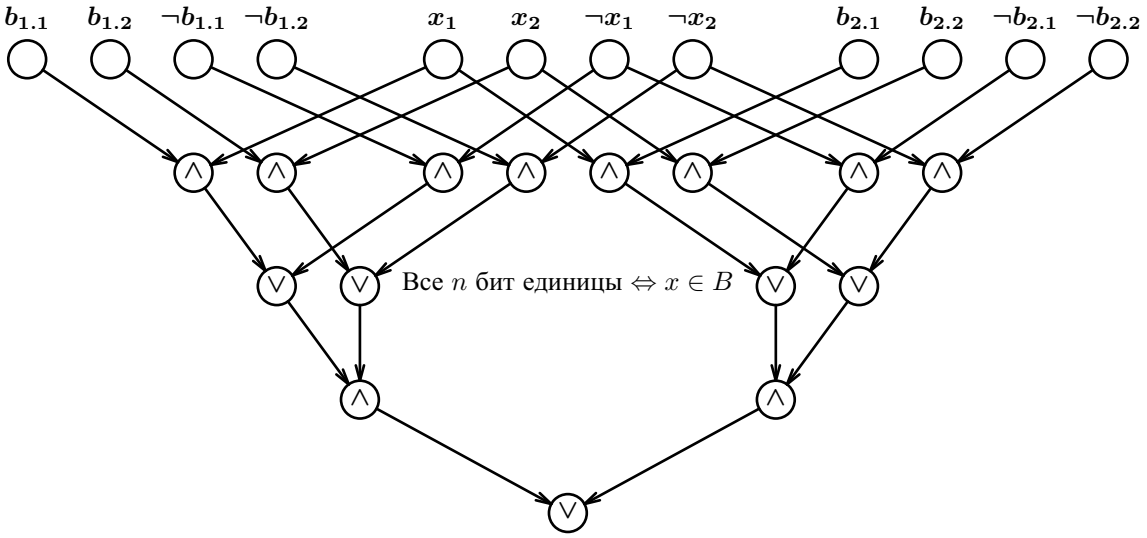

Рис. 2.5: Пример — $n = 2$, $|B| = 2$.

Определение 54. Пусть M — конечный моноид. Тогда ветвящаяся программа P на M с n входами — это последовательность $I_1, \dots, I_l$ троек $\langle i, g_0, g_1\rangle$, где $1 \leqslant i \leqslant n$ и $g_0, g_1 \in M$, плюс *допускающее множество* $A \subseteq M$. Эти тройки называем *инструкциями* программы.

Определение 55. Ветвящаяся программа P на моноиде M с n входами *вычисляет* булеву функцию $f : \{0,1\}^n \to \{0,1\}$ так. Пусть вход $x = (x_1, \dots, x_n) \in \{0,1\}^n$. Результат выполнения каждой инструкции $I = \langle i, g_0, g_1\rangle$ равен (обозначим его $I_i(x)$) g_{x_i} (то есть g_0, если $x_i = 0$, и g_1, если $x_i = 1$). Результат вычисления $P(x) = \prod_{j=1}^{l} I_j(x)$. $f(x) = 1 \Leftrightarrow P(x) \in A$.

Замечание. Перестановочные ветвящиеся программы — частный случай ветвящихся программ на моноидах. Ранее мы рассматривали программу как множество троек $\langle i, \sigma, \tau\rangle$, где σ, τ — две перестановки, а результат вычисления программы — композиция всех перестановок.

Определение 56. *Длина* ветвящейся программы на моноиде — число её инструкций (l в наших обозначениях).

2.2. Задача — посчитать произведение элементов моноида/группы.

Пусть M — конечный моноид. Обозначим $PROD(M)$ задачу: по входу $x_1, x_2, \ldots, x_n \in M$ посчитать $x_1 \cdot x_2 \cdot \ldots \cdot x_n$. Посчитать выход ветвящейся программы на моноиде — частный случай этой задачи.

Вычислять произведение элементов моноида будем с помощью булевых схем. Для этого кодируем элементы M двоичными последовательностями. Пусть $M = \{y_1, y_2, \ldots, y_m\}$ — как-то упорядочим его элементы. Поставим в соответствие каждому y_i двоичную строку 1^i0^{m-i}. Теперь, если мы хотим подать на вход булевой схеме последовательность элементов M $x_1, x_2, \ldots, x_n$, то просто закодируем их вышеописанным способом.

Определение 57. *Оракульный вход* для $PROD(M)$ — связка, выдающая по элементам $m_1, m_2, \ldots, m_s \in M$ их произведение $m_1 \cdot m_2 \cdot \ldots \cdot m_s$. В данной главе будем считать, что если подать оракулу некорректный набор бит (не элементы M, а просто какую-то последовательность нулей и единиц), то выходом будут все нули (ни один элемент M так не кодируется).

Лемма 15. Если M — конечный моноид левых нулей, то $PROD(M)$ решается с помощью схем из AC^0.

Доказательство.

□ Пусть $x_1, x_2, \ldots, x_n \in M$. Так как M — моноид левых нулей, то $x_1 \cdot x_2 \cdot \ldots \cdot x_n = x_t$, где x_t — не единица в M и $x_1, x_2, \ldots, x_{t-1}$ — единицы (если таковой существует, иначе — единица). Построим вспомогательную схему, которая по входу $x \in M$ даёт 1 тогда и только тогда, когда x — единица. Например, можно упорядочить элементы M так, чтобы единица была первой ($M = \{y_1, y_2, \ldots, y_m\}$) и поставить в соответствие каждому y_i строку 1^i0^{m-i}. Тогда единица будет закодирована как 10^{m-1}. Пример — на рисунке 2.6.

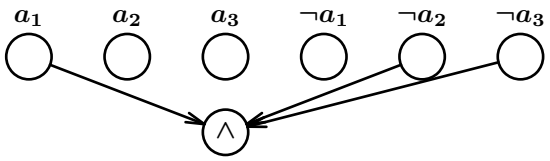

Рис. 2.6: Так будет выглядеть схема, если x кодируется тремя битами a_1, a_2, a_3. Для произвольного m берём конъюнкцию $a_1, \neg a_2, \neg a_3, \ldots, \neg a_m$. Назовём эту схему «$e$».

Применим эту схему ко всем x_i и обозначим полученные функции φ_i. Затем для каждого i посчитаем конъюнкцию $\neg\varphi_i, \varphi_1, \varphi_2, \ldots, \varphi_{i-1}$. Получим единицу тогда и только тогда, когда φ_i — не единица в M, а все $\varphi_1, \varphi_2, \ldots, \varphi_{i-1}$ — единицы. Возьмём побитовую конъюнкцию результата и x_i ($\forall\ i$ получим x_i, если x_i первый неединичный элемент, и все нули иначе), а затем дизъюнкцию по всем i. Это и будет выход схемы для $PROD(M)$. На рисунке 2.7 — эта схема, если $n = 3$ и все x_i кодируются тремя битами. Вновь по лемме 13 можно использовать как связку, так и её отрицание.

На самом деле эта схема немного неправильная. Она не учитывает случай, когда все x_i — единицы, в этом случае она выдаст все нули, а не единицу. Обозначим T конъюнкцию всех φ_i (1 если и только если $\forall\ i : x_i = e$). Надо взять конъюнкцию каждого бита e с T, а потом

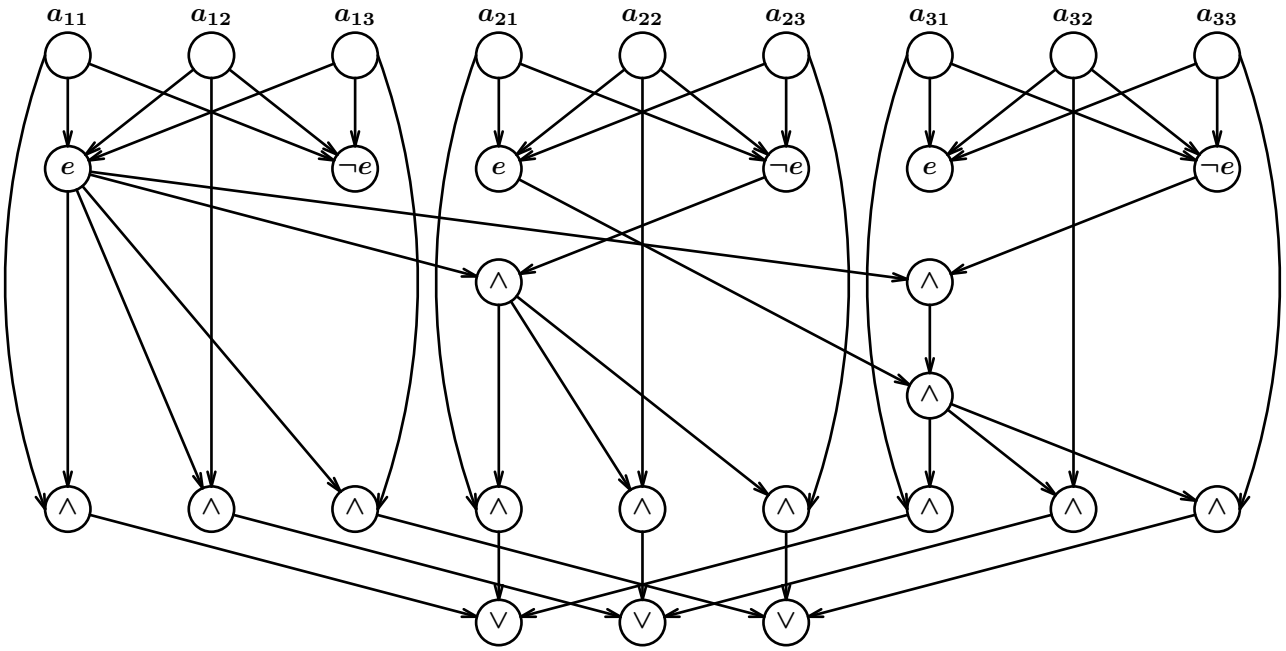

Рис. 2.7: Схема для случая, когда $n = 3$ и все x_i кодируются тремя битами. Обозначим $x_1 = a_{11}a_{12}a_{13}$, $x_2 = a_{21}a_{22}a_{23}$, $x_3 = a_{31}a_{32}a_{33}$.

потом побитовую дизъюнкцию результата и выхода построенной ранее схемы. Если среди x_i были неединичные элементы, выход схемы не изменится; если же вход — все единицы, то выходом тоже будет единица. ■

Лемма 16. Если G — конечная разрешимая группа, то $PROD(G)$ решается с помощью схем из ACC^0.

Доказательство.

□ Так как G конечна, то у неё есть композиционный ряд $\{e\} = G_s \lhd \ldots \lhd G_1 \lhd G_0 = G$. По лемме 12 $\forall\ i : G_{i+1}/G_i$ — циклическая группа простого порядка. Будем строить нужную схему индукцией по s. $PROD(G_s)$ решается тривиально — по любому входу $x_1, x_2, \ldots, x_n \in G_s$ возвращаем e. Теперь осталось решить $PROD(G)$, имея оракульные схемы для $PROD(G_1)$.

Пусть $|G/G_1| = m$, $|G| = N$, $A \in G/G_1$ и $G/G_1 = \{A^0, A^1, \ldots, A^{m-1}\}$, $a \in G$ — такой, что $A = aG_1 = \{ax \mid x \in G_1\}$. Тогда (так как G_1 нормальна) $\forall i : A^i = \underbrace{(aG_1)(aG_1)\ldots(aG_1)}_{i \text{ раз}} = aG_1G_1a\underbrace{(aG_1)(aG_1)\ldots(aG_1)}_{i-2 \text{ раза}} = aG_1a\underbrace{(aG_1)(aG_1)\ldots(aG_1)}_{i-2 \text{ раза}} = a^2G_1\underbrace{(aG_1)(aG_1)\ldots(aG_1)}_{i-2 \text{ раза}} = \ldots = a^iG_1$. Пусть $x_1, x_2, \ldots, x_n \in G$ — вход, нужно посчитать $x_1x_2\ldots x_n$. Тогда $\forall\ i : x_i = a^{k_i}b_i$ для некоторых $k_i \in \{0, 1, \ldots, m-1\}$ и $b_i \in G_1$. Обозначим $y_i = a^{k_1+\ldots+k_i}b_ia^{-(k_1+\ldots+k_i)} \in G_1$.

Так как $\Big(\prod\limits_{i=1}^{n} y_i\Big)a^{k_1+\ldots+k_n} = a^{k_1}b_1a^{-k_1}a^{k_1+k_2}b_2a^{-k_1-k_2}\ldots a^{k_1+\ldots+k_n}b_na^{-k_1-\ldots-k_n}a^{k_1+\ldots+k_n}$ $= a^{k_1}b_1a^{k_2}b_2\ldots a^{k_n}b_n = \prod\limits_{i=1}^{n} x_i$, достаточно научиться считать y_i (а для их произведения $y_1\ldots y_n$ существует оракульная схема), а потом домножить $y_1\ldots y_n$ на $a^{k_1+\ldots+k_n}$. Схема для произведения двух элементов группы существует по лемме 14 — при заданной кодировке элементов G найдётся булева схема от $2 \cdot |G|$ аргументов, возвращающая произведение соответствующих

элементов G, или все нули при некорректном входе. Потребуются такие вспомогательные схемы — по входу x_i посчитать a^{k_i} и b_i, и по входу $a^{k_1}, a^{k_2}, \ldots, a^{k_i}$ посчитать $a^{k_1+k_2+\ldots+k_i}$ и $a^{-k_1-k_2-\ldots-k_i}$.

Сначала построим a^{k_i} и b_i по входу x_i. Пусть $G_1 = \{g_1, g_2, \ldots, g_{N/m}\}$ — произвольным образом упорядочим элементы G_1. Сначала вычислим всевозможные произведения $a^{j_1} g_{j_2}$, где $0 \leqslant j_1 < m$, $1 \leqslant j_2 \leqslant N/m$. Потом применим к каждому из них схему, которую обозначим B_i — побитовая конъюнкция с x_i. Возьмём конъюнкции всех N бит для каждого применения B_i (обозначим эту схему D). $D = 1$ тогда и только тогда, когда $x_i = a^{j_1} g_{j_2}$.

Пусть K_{j_1} — конъюнкция всех D, соответствующих одному a^{j_1} ($0 \leqslant j_1 < m$). $K_{j_1} = 1$ тогда и только тогда, когда $x_i = a^{j_1} g_{j_2}$ для некоторого $j_2 : 1 \leqslant j_2 \leqslant N/m$. То есть надо $\forall\, j_1 : 0 \leqslant j_1 < m$ посчитать конъюнкцию каждого бита x_i с K_{j_1}; a_{k_i} — побитовая дизъюнкция всех таких штук.

Обозначим $\forall j_2 : 1 \leqslant j_2 \leqslant G/m$ и для каждого D (соответствующего a^{j_1} и g_{j_2}): T_{j_2} — конъюнкция всех бит g_{j_2} с D. $T_{j_2} = g_{j_2}$, если $x_i = a^{j_1} g_{j_2}$ для некоторого j_1; и все нули — иначе. b_i — побитовая дизъюнкция T_{j_2} по всем j_2. Схема — на рисунке 2.8 (опускаем часть стрелок для наглядности).

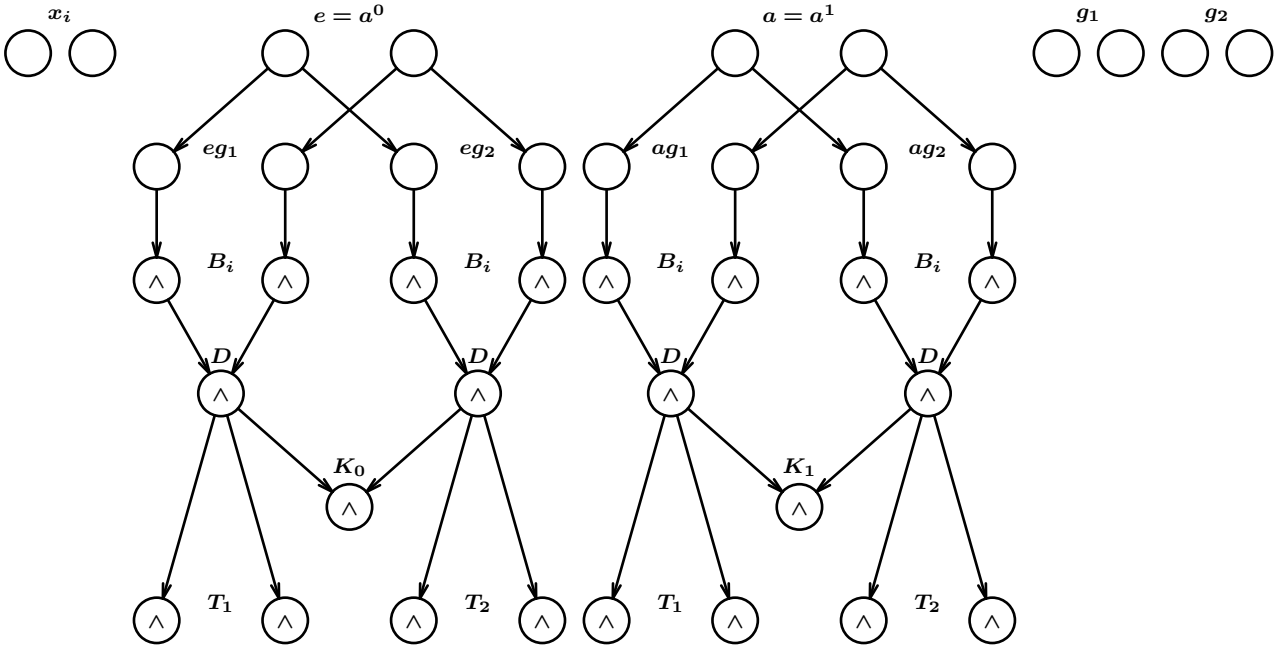

Рис. 2.8: Пример для $N = 2$ и $m = 2$.

Теперь посчитаем $a^{k_1+k_2+\ldots+k_i}$ по входу $a^{k_1}, a^{k_2}, \ldots, a^{k_i}$.

Заметим, что $a^{k_1+k_2+\ldots+k_i} = a^{(k_1+k_2+\ldots+k_i) \pmod{\operatorname{ord} a}}$. По лемме 14 существует схема, по всякому a^j возвращающая j. Пусть $\bar{n}$ — строка из последовательно записанных $k_1, k_2, \ldots, k_i$. Будем действовать аналогично тому, как если бы нужно было построить схему, которая бы вычисляла число по модулю константы. Входом нашей схемы будут строки $\bar{n}$, $\bar{n}\,1$ (конкатенация $\bar{n}$ и 1), …, $\bar{n}\,1^{\operatorname{ord} a - 1}$. К каждой из них применяем связку $MOD_{\operatorname{ord} a}$. $\forall\, j, 0 \leqslant j < \operatorname{ord} a$: берём побитовую конъюнкцию $MOD_{\operatorname{ord} a}$ с j. И берём побитовую дизъюнкцию по всем j.

Получаем $(k_1 + k_2 + \ldots + k_i) \pmod{\operatorname{ord} a}$. И применяем схему, которая по $j : 0 \leqslant j < \operatorname{ord} a$ возвращает a^j. Такая схема существует по лемме 14.

Аналогично строится схема для $a^{-k_1-k_2\ldots-k_i}$. Надо сначала по входу $a^{k_1}, a^{k_2}, \ldots, a^{k_i}$ получить для всех j: $1 \leqslant j \leqslant k_i$ $a^{\operatorname{ord} a - k_j}$, потому что $a^{-k_j} = a^{\operatorname{ord} a - k_j}$. Или — хотя бы строки, содержащие $\operatorname{ord} a - k_j$ единиц (их конкатенацию возьмём в качестве $\bar{n}$). По лемме 14 существует схема, которая по a^λ, где $0 \leqslant \lambda < m$, возвращает строку $\underbrace{00\ldots0}_{\lambda}\underbrace{11\ldots1}_{\operatorname{ord} a-\lambda}$ длины $\operatorname{ord} a$, содержащую нужное число единиц. ■

Лемма 17. Пусть M — конечный моноид, $M = N_1 \cup N_2$, N_1 и N_2 — собственные подмоноиды, $N_1/\{1\}$ — левый идеал в M. Тогда $PROD(M)$ решается с помощью схем из AC^0 со оракульными входами для $PROD(N_1)$ и $PROD(N_2)$.

Доказательство.

□ Пусть $x_1, x_2, \ldots, x_n \in M$ — вход. Попытаемся подобрать такие $x'_1, x'_2, \ldots, x'_n, x'_{n+1}$, что $x_1x_2\ldots x_n = x'_1x'_2\ldots x'_nx'_{n+1}$, $\forall\ i\ (1 \leqslant i \leqslant n) : x_i \in N_1$ и $x'_{n+1} \in N_2$.

Пусть интервал $[a, b]$ $(1 \leqslant a \leqslant b \leqslant n)$ такой, что $\forall\ i\ (a \leqslant i \leqslant b) : x_i \in N_2$. Тогда существует оракульная схема, вычисляющая $x_{a,b} = x_ax_{a+1}\ldots x_b$ (ей подаются на вход $x_a, x_{a+1}, \ldots, x_b$). Назовём её *разрешённой*, если $[a, b]$ имеет наибольшую длину, то есть $x_{a-1} \notin N_2$ (или $a = 1$) и $x_{b+1} \notin N_2$ (или $b = n$), а сам интервал *разрешённым*.

Обозначим $x_{n+1} = 1$ (элемент M). Пусть при $1 \leqslant i \leqslant n + 1$ $x'_i = 1$, если i принадлежит разрешённому интервалу; $x'_i = x_{a,b}x_i$, если существует такой разрешённый интервал $[a, b]$, что $i = b + 1$; и $x'_i = x_i$, иначе. Так как $N_1/\{1\}$ — левый идеал в M, то при $x'_i = x_{a,b}x_i$ (умножаем элемент N_2 на элемент N_1) $x'_i \in N_1/\{1\}$.

Нужно построить схему $C_1(x)$, которая по входу $x \in M$ даёт 1, если $x \in N_1$, и 0 иначе. Используем лемму 14 о том, что для всякой булевой функции существует вычисляющая её схема. Пусть мы кодируем элементы M, как и раньше — как-то упорядочиваем его элементы $M = \{m_1, m_2, \ldots, m_t\}$ и ставим в соответствие каждому m_i 1^i0^{t-i}. Обозначим элементы N_1 как $\{m_{I_1}, m_{I_2}, \ldots\}$; $\forall\ j : m_{I_j} = m_{j.1}m_{j.2}\ldots m_{j.t}$. Тогда $f(a_1, a_2, \ldots, a_t) = \Big((a_1 = m_{1.1}) \wedge (a_2 = m_{1.2}) \wedge \ldots \wedge (a_t = m_{1.t})\Big) \vee \Big((a_1 = m_{2.1}) \wedge (a_2 = m_{2.2}) \wedge \ldots \wedge (a_t = m_{2.t})\Big) \vee \ldots$ — и есть функция проверки принадлежности элемента M к N_1. По такой функции несложно построить схему, вычисляющую её, причём её размер зависит только от размеров M и N_1, то есть конечен.

Теперь по любому интервалу $[a, b]$ построим схему, проверяющую, является ли он разрешённым. Вход — $\{x_a, x_{a+1}, \ldots, x_b\}$. Выход — 1, если $[a, b]$ разрешённый, и 0 иначе. Просто берём конъюнкцию выходов $C_1(x_a), C_1(x_{a+1}), \ldots, C_1(x_b)$. Назовём эту схему $C_{a,b}$.

Следующая вспомогательная схема — назовём её $P_{a,b}$ — по входу $x_1, x_2, \ldots, x_i$ даёт $x_{a,b}$, где $[a, b]$ — такой разрешённый интервал, что $i = b + 1$ (или все нули, если такой $[a, b]$ не существует). Существует оракульная схема для вычисления произведения $x_1x_2\ldots x_j$. Берём $\forall\ j : 1 \leqslant j \leqslant i - 1$ побитовую конъюнкцию $x_1x_2\ldots x_j$ и $C_{j,i-1}$, а потом — побитовую конъюнкцию с $\neg C_{j-1,i-1}$ (кроме случая $j = 1$). Не все нули будут только если $[j, i - 1]$ — разрешённый. Побитовая дизъюнкция по всем j результатов — это выход схемы.

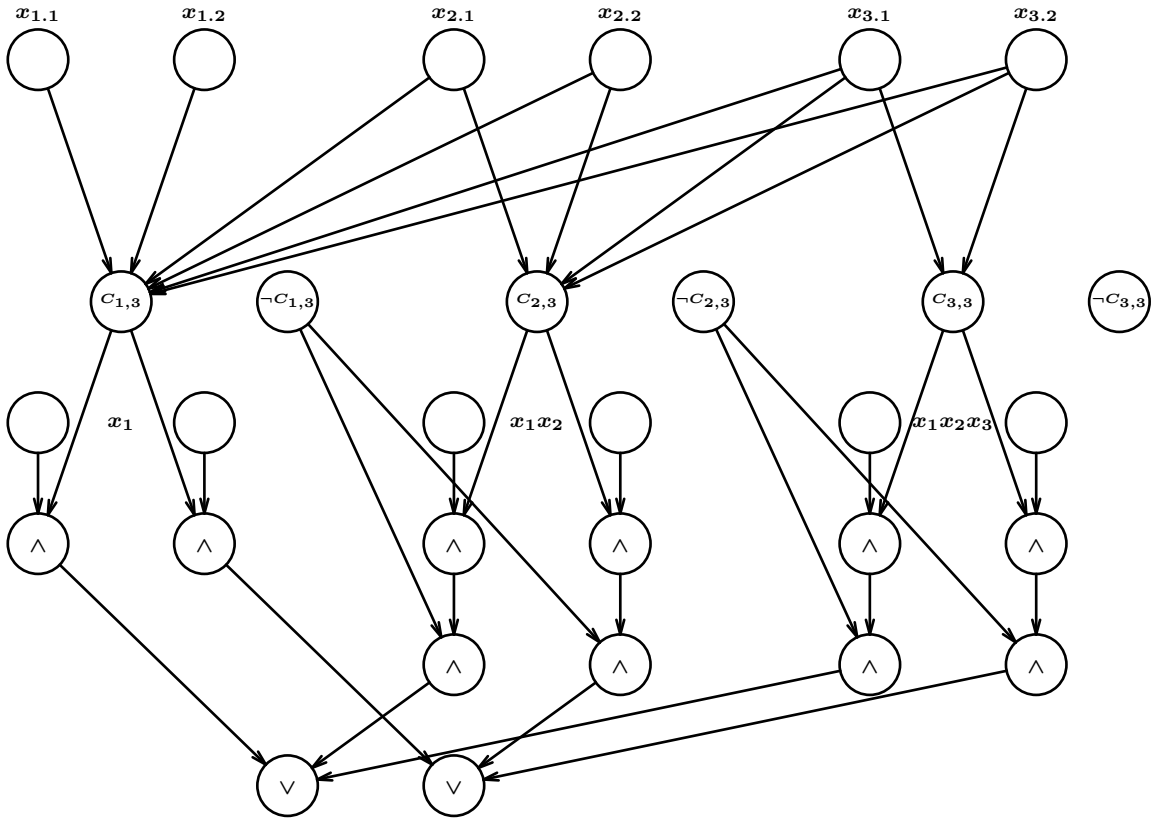

Рис. 2.9: Пример — $i = 4, t = 2$. Для наглядности опускаем некоторые стрелки, например ведущие в оракульные схемы для произведения.

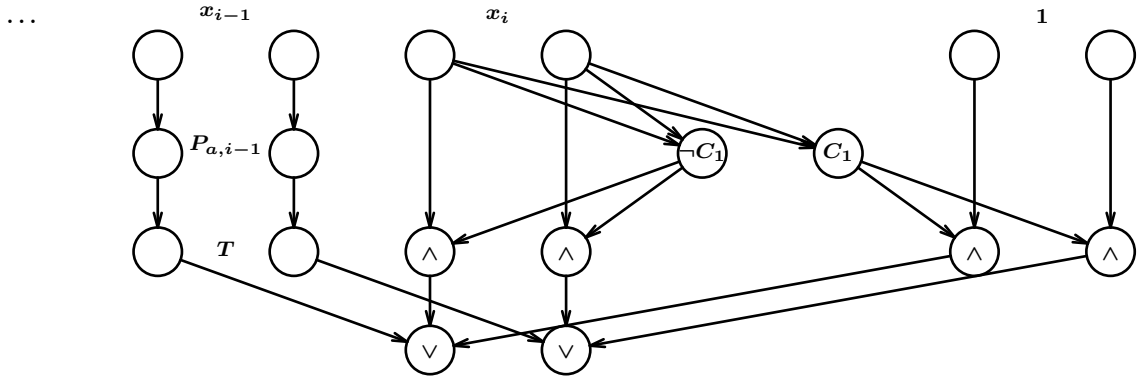

Рис. 2.10: Пример — $t = 2$, i — любое число, превосходящее 1.

На рисунке 2.9 — схема для $P_{a,b}$. А на рисунке 2.10 — схема для x'_i. Построим x'_i. Пусть $T(x, y)$ — схема, вычисляющая xy — произведение двух элементов M; даёт все нули при некорректном входе. Тогда x'_i — побитовая дизъюнкция $T(P_{a,i-1})$, конъюнкции x_i с $\neg C_1(x_i)$ и конъюнкции 1 (нейтрального элемента моноида) с $C_1(x_i)$. Осталось вычислить $x'_1x'_2 \dots x'_n$ оракульной схемой и умножить на x'_{n+1} схемой T. ■

Лемма 18. Пусть M — конечный непериодический моноид. Тогда $PROD(M)$ решается с помощью схем из AC^0.

Доказательство.

□ Следует из теоремы 4 статьи [4]. ■

Глава 3. Ветвящиеся программы малой ширины.

Назовём $k - PBP$ — функции, вычислимые перестановочными ветвящимися программами ширины k и полиномиальной длины. Хотим понять, как эти классы вкладываются в $AC^0, AC^0[m]$, где $m \in \mathbb{N}$. Также будем изучать OBP — *забывающие ветвящиеся программы* — детерминированные ветвящиеся программы, вершины которых можно разбить на уровни так, чтобы из каждого i-го уровня рёбра вели только в $(i+1)$-й и на каждом уровне все вершины соответствовали одной переменной.

Перестановочные программы — это ветвящиеся программы на моноидах S_k. А забывающим ветвящимся программам соответствуют моноиды всюду определённых функций из $\{1, 2, \ldots, k\}$ в себя; необязательно биекций. Будем обозначать эти моноиды F_k.

Построим таблицу — для классов булевых функций $k - PBP$ и $k - OBP$ (при $k \in \{2, 3, 4\}$), какому моноиду соответствуют программы, вычисляющие эти функции, и в какой класс функций, вычислимых схемами из ACC^0, $k - PBP$ ($k - OBP$) вкладывается.

Тип программы	Моноид	В какой класс булевых схем вкладывается
$2 - PBP$	S_2	$AC^0[2]$
$3 - PBP$	S_3	$AC^0[6]$
$4 - PBP$	S_4	$AC^0[12]$
$2 - OBP$	F_2	$AC^0[2]$
$3 - OBP$	F_3	$AC^0[6]$
$4 - OBP$	F_4	$AC^0[12]$

А в главе 1 доказано, что $NC^1 \subseteq 5 - PBP$. На самом деле равенство: $5 - PBP = NC^1$.

3.1. Связи между классами ветвящихся программ

Лемма 19. $\forall\ k \in \mathbb{N} : k - PBP \subseteq k - OBP$.

Доказательство.

□ Пусть $f \in k - PBP$. Тогда существует перестановочная ветвящаяся программа P, вычисляющая f. Но любая перестановочная ветвящаяся программа является забывающей (любая перестановка из S_k является всюду определённой функцией из $[k]$ в $[k]$), то есть P — забывающая ветвящаяся программа ширины k, вычисляющая f и $f \in k - OBP$. ■

Лемма 20. $\forall\ k \in \mathbb{N} : k - PBP \subseteq (k+1) - PBP$ и $k - OBP \subseteq (k+1) - OBP$.

Доказательство.

□ Пусть $f \in k - PBP$. Тогда существует перестановочная ветвящаяся программа P ширины k, вычисляющая f. Пусть $\sigma_1, \ldots, \sigma_t, \tau_1, \ldots, \tau_t$ — перестановки; σ_i соответствует переходу i-й инструкции по 1-рёбрам, а τ_i — по 0-рёбрам. Поставим в соответствие каждой $\sigma_i = \left(\begin{smallmatrix} 1 & \cdots & k \\ \sigma_{i.1} & \cdots & \sigma_{i.k} \end{smallmatrix}\right)$ перестановку $\sigma'_i = \left(\begin{smallmatrix} 1 & \cdots & k & k+1 \\ \sigma_{i.1} & \cdots & \sigma_{i.k} & k+1 \end{smallmatrix}\right)$; аналогично для τ. Для каждого входа

$x = (x_1, \ldots, x_n)$ программы P $f(x) = P(x)$; $P(x) = 0$ тогда и только тогда, когда $e = \pi_1 \ldots \pi_t$, где «π» означает «σ или τ». Заменим в этом произведении каждую σ_i на σ_i', а τ_i на τ_i'. Получим новую программу $P'(x)$ ширины $k+1$. Пусть $\rho = \pi_1 \ldots \pi_t = \left(\begin{smallmatrix} 1 & \ldots & k \\ \rho_1 & \ldots & \rho_k \end{smallmatrix}\right)$, $\rho' = \pi_1' \ldots \pi_t' = \left(\begin{smallmatrix} 1 & \ldots & k & k+1 \\ \rho_1 & \ldots & \rho_k & k+1 \end{smallmatrix}\right)$. Тогда $\rho = e \Leftrightarrow \rho' = e$; $f(x) = 0 \Leftrightarrow \rho = e$. Отсюда $f(x) = 0 \Leftrightarrow \rho' = e$, то есть P' вычисляет f. Рассуждение для моноида биекций $[k] \to [k]$ полностью аналогично. ■

3.2. Перестановочные ветвящиеся программы.

3.2.1. Перестановочные ветвящиеся программы ширины 2

$S_2 = \left\{ \left(\begin{smallmatrix} 1 & 2 \\ 1 & 2 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 2 \\ 2 & 1 \end{smallmatrix}\right) \right\}$ (обозначим $\{e, a\}$). Так как $a \cdot a = e$, то $a = a^{-1}$; $e = e^{-1}$. Значит, S_2 — группа. $\forall\, g, h \in S_2$ $[g, h] = g^{-1}h^{-1}gh = ghgh$:

	e	a
e	$eeee = e$	$eaea = e$
a	$aeae = e$	$aaaa = e$

$S_2' = \{e\}$ и S_2 разрешимая группа (очевидно, $\{e\}$ — нормальная подгруппа). Значит, S_2 подходит под условие леммы 16. Заметим, что в доказательстве этой леммы используются только связки вида $MOD_{\operatorname{ord} a_i}$, где $a_i \in G_i$. $\operatorname{ord} e = 1$; $\operatorname{ord} a = 2$. Следовательно, $2 - PBP \subseteq AC^0[2]$.

3.2.2. Перестановочные ветвящиеся программы ширины 3

$S_3 = \left\{ \left(\begin{smallmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{smallmatrix}\right) \right\} = \Big\{e, (123), (132), \tau_{23}, \tau_{13}, \tau_{12}\Big\}$.
Построим таблицу для произведения этих элементов:

	e	τ_{12}	τ_{13}	τ_{23}	(123)	(132)
e	e	τ_{12}	τ_{13}	τ_{23}	(123)	(132)
τ_{12}	τ_{12}	e	(123)	(132)	τ_{13}	τ_{23}
τ_{13}	τ_{13}	(132)	e	(123)	τ_{23}	τ_{12}
τ_{23}	τ_{23}	(123)	(132)	e	τ_{12}	τ_{13}
(123)	(123)	τ_{23}	τ_{12}	τ_{13}	(132)	e
(132)	(132)	τ_{13}	τ_{23}	τ_{12}	e	(123)

Значит, обратные элементы:

g	e	τ_{12}	τ_{13}	τ_{23}	(123)	(132)
g^{-1}	e	τ_{12}	τ_{13}	τ_{23}	(132)	(123)

Таблица для коммутаторов элементов S_3:

	e	τ_{12}	τ_{13}	τ_{23}	(123)	(132)
e	e	e	e	e	e	e
τ_{12}	e	e	(123)	(132)	(132)	(123)
τ_{13}	e	(132)	e	(123)	(132)	(123)
τ_{23}	e	(123)	(132)	e	(132)	(123)
(123)	e	(123)	(123)	(123)	e	e
(132)	e	(132)	(132)	(132)	e	e

$S_3' = \Big\{e, (123), (132)\Big\}$. Видно, что $(S_3')' = ((S_3')')' = \ldots = \{e\}$. Значит, S_3 разрешима и применима лемма 16. Причём по теореме Лагранжа S_3' — наибольшая из подгрупп S_3 (3 — самый большой множитель 6), а $(S_3')'$ — наибольшая из подгрупп S_3'. По лемме 11 коммутатор является нормальной подгруппой. Значит, коммутаторы являются членами композиционного ряда. Все возможные порядки элементов S_3 — $\{1, 2, 3\}$, следовательно, используем связки MOD_2 и MOD_3. На самом деле, достаточно связки MOD_6 — по связке MOD_6 несложно построить связку MOD_3. $\forall\ x \in \mathbb{N}\ MOD_3(x) = MOD_6(x) \vee MOD_6(x+3)$. Аналогично для MOD_2. В общем случае: если нужно использовать связки $MOD_{n_1}, MOD_{n_2}, \ldots, MOD_{n_s}$, то достаточно использовать связку $MOD_{\text{НОК}(n_1,\ldots,n_s)}$, где НОК — наименьшее общее кратное. Значит, $3 - PBP \subseteq AC^0[6]$.

3.2.3. Перестановочные ветвящиеся программы ширины 4

Проверим, является ли S_4 разрешимой группой. Построим таблицу коммутаторов:

	e	$\begin{pmatrix}1&2&3&4\\1&2&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&3&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$
e	e	e	e	e	e	e	e	e
$\begin{pmatrix}1&2&3&4\\1&2&4&3\end{pmatrix}$	e	e	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	e	e
$\begin{pmatrix}1&2&3&4\\1&3&2&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	e	e	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	e	e	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&4&3&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&1&3&4\end{pmatrix}$	e	e	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	e	e
$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	e	e	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	e	e
$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&3&4&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&4&1&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&1&4&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&2&1&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	e
$\begin{pmatrix}1&2&3&4\\3&4&2&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	e
$\begin{pmatrix}1&2&3&4\\4&1&2&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\4&2&3&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\4&3&1&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	e
$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	e

	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$

e	e	e	e	e	e	e	e	e
$\begin{pmatrix}1&2&3&4\\1&2&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&3&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&4&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&1&3&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&3&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&4&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&1&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&2&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	e
$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&4&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\4&1&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	e
$\begin{pmatrix}1&2&3&4\\4&2&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\4&3&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$

	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$
e	e	e	e	e	e	e	e	e
$\begin{pmatrix}1&2&3&4\\1&2&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&3&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	e
$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\1&4&3&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&1&3&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	e	e	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	e	e
$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&3&4&1\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\2&4&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	e
$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\2&4&3&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$
$\begin{pmatrix}1&2&3&4\\3&1&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&1&3&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&1&2&4\end{pmatrix}$	e
$\begin{pmatrix}1&2&3&4\\3&2&1&4\end{pmatrix}$	e	$\begin{pmatrix}1&2&3&4\\2&3&1&4\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&2&4&1\end{pmatrix}$	$\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}$

$\left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&3&1&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\1&4&2&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e
$\left(\begin{smallmatrix}1&2&3&4\\3&4&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&4&3&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&1&2&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\1&4&2&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&1&2&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\3&1&2&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&2&1&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\1&4&2&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&3&1&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&3&1&4\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&2&1&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\1&3&4&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&4&3&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\4&2&1&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&1&2&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&2&3&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&2&1&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&4&3&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&4&3&1\end{smallmatrix}\right)$	e
$\left(\begin{smallmatrix}1&2&3&4\\4&3&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\1&3&4&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&2&1&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&3&1&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e

$S'_4 = \{e, \left(\begin{smallmatrix}1&2&3&4\\1&3&4&2\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\1&4&2&3\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\4&2&1&3\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\2&4&3&1\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\2&3&1&4\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\3&1&2&4\end{smallmatrix}\right),$
$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)\}$. Таблица коммутаторов для S'_4:

	$\left(\begin{smallmatrix}1&2&3&4\\4&2&1&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&3&1&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\1&3&4&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\1&4&2&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&2&1&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\2&3&1&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$
e	e	e	e	e	e	e
$\left(\begin{smallmatrix}1&2&3&4\\1&3&4&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	e	e
$\left(\begin{smallmatrix}1&2&3&4\\1&4&2&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	e	e
$\left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\2&4&3&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e	e	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	e	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\3&1&2&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$

	$\left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&4&3&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&1&2&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&2&1&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\2&3&1&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$
e	e	e	e	e	e	e
$\left(\begin{smallmatrix}1&2&3&4\\1&3&4&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\1&4&2&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\3&2&4&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\2&4&3&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e
$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	e	e	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e	e	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\3&1&2&4\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	e	$\left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right)$	e

Таким образом, $(S'_4)' = \{e, \left(\begin{smallmatrix}1&2&3&4\\2&1&4&3\end{smallmatrix}\right), \left(\begin{smallmatrix}1&2&3&4\\3&4&1&2\end{smallmatrix}\right)\left(\begin{smallmatrix}1&2&3&4\\4&3&2&1\end{smallmatrix}\right)\}$. Для любых двух элементов $(S'_4)'$ коммутатор будет равен e. Таким образом, $((S'_4)')' = \{e\}$ и S_4 разрешима. Так же, как для S_3, по теореме Лагранжа, S'_4 — наибольшая подгруппа для S_4, а $((S'_4)')'$ — для $(S'_4)'$ (12 —

наибольший делитель 24, 1 — для 4). Надо проверить, нет ли в S_4' подгруппы размера 6. Предположим, что такая группа существует (и докажем обратное). Обозначим эту группу H. H содержит хотя бы 2 цикла длины 3. Обозначим первый из них $\left(\begin{smallmatrix} i & j & k & l \\ j & k & i & l \end{smallmatrix}\right)$. Второй может либо иметь вид $\left(\begin{smallmatrix} i & j & k & l \\ k & i & j & l \end{smallmatrix}\right)$, то есть оставлять на месте ту же цифру, либо другую. Рассмотрим только случаи $\left(\begin{smallmatrix} i & j & k & l \\ i & k & l & j \end{smallmatrix}\right)$ и $\left(\begin{smallmatrix} i & j & k & l \\ i & l & j & k \end{smallmatrix}\right)$, остальные аналогичны.

Обозначим ① = $\left(\begin{smallmatrix} i & j & k & l \\ j & k & i & l \end{smallmatrix}\right)$, ② = $\left(\begin{smallmatrix} i & j & k & l \\ k & i & j & l \end{smallmatrix}\right)$. ① · ② = $\left(\begin{smallmatrix} i & j & k & l \\ i & j & k & l \end{smallmatrix}\right) = e$. ② · ① = $\left(\begin{smallmatrix} i & j & k & l \\ i & j & k & l \end{smallmatrix}\right) = e$. Получаем группу всего из двух элементов, а не из шести.

Обозначим ① = $\left(\begin{smallmatrix} i & j & k & l \\ j & k & i & l \end{smallmatrix}\right)$, ② = $\left(\begin{smallmatrix} i & j & k & l \\ i & k & l & j \end{smallmatrix}\right)$. ① · ② = $\left(\begin{smallmatrix} i & j & k & l \\ k & l & i & j \end{smallmatrix}\right)$ = ③. ① · ③ = $\left(\begin{smallmatrix} i & j & k & l \\ l & i & k & j \end{smallmatrix}\right)$ = ④. ③ · ④ = $\left(\begin{smallmatrix} i & j & k & l \\ k & j & l & i \end{smallmatrix}\right)$ = ⑤. ① · ⑤ = $\left(\begin{smallmatrix} i & j & k & l \\ j & l & k & i \end{smallmatrix}\right)$ = ⑥. Получаем, что H уже содержит 6 различных элементов, не включая e. Значит, $|H| = 12$.

Обозначим ① = $\left(\begin{smallmatrix} i & j & k & l \\ j & k & i & l \end{smallmatrix}\right)$, ② = $\left(\begin{smallmatrix} i & j & k & l \\ i & l & j & k \end{smallmatrix}\right)$. ① · ② = $\left(\begin{smallmatrix} i & j & k & l \\ l & j & i & k \end{smallmatrix}\right)$ = ③. ① · ③ = $\left(\begin{smallmatrix} i & j & k & l \\ j & i & l & k \end{smallmatrix}\right)$ = ④. ③ · ④ = $\left(\begin{smallmatrix} i & j & k & l \\ k & i & j & l \end{smallmatrix}\right)$ = ⑤. ④ · ⑤ = $\left(\begin{smallmatrix} i & j & k & l \\ i & k & j & l \end{smallmatrix}\right)$ = ⑥. Получаем, что H уже содержит 6 различных элементов, не включая e. Значит, $|H| = 12$. Получаем противоречие с тем, что $|H| = 6$, значит, $(S_4')'$ — наибольшая подгруппа S_4'.

По лемме 11 коммутатор является нормальной подгруппой. Следовательно, коммутаторы являются членами композиционного ряда. Порядки элементов S_4 — $1, 2, 3, 4$. Значит, используем связку MOD_{12}. $4 - PBP \subseteq AC^0[12]$.

3.3. Забывающие ветвящиеся программы.

3.3.1. Забывающие ветвящиеся программы ширины 2

Элементы F_2 выглядят так:

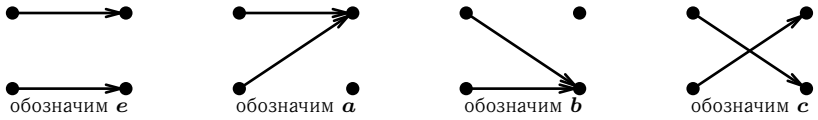

Рис. 3.1: Всевозможные функции f: $\{1, 2\} \to \{1, 2\}$

«Таблица умножения»:

	e	a	b	c
e	e	a	b	c
a	a	a	b	b
b	b	a	b	a
c	c	a	b	e

Таблица обратных значений:

g	e	a	b	c
g^{-1}	e	-	-	c

Таким образом, F_2 — даже не группа. Обозначим $N_1 = \{e, a, b\}$, $N_2 = \{e, c\}$, то N_1 и N_2 моноиды и $N_1 \setminus \{e\}$ — левый идеал в F_2. Применима лемма 17. $N_2 = S_2$, то есть оракульная схема для $PROD(N_2)$ принадлежит $AC^0[2]$. Заметим, что $\forall\ x, y \in N_1, y \neq e$: $xy = y$, то есть N_1 — моноид правых нулей. Если сформулировать аналог леммы 15 для моноидов правых нулей, то получим, что $PROD(N_1)$ разрешима схемами из AC^0. Значит, $2 - OBP \subseteq AC^0[2]$.

Здесь заметим, что элементы F_2 кодируются как 1^i0^{4-i}, где $i \in \{1,2,3,4\}$. А оракульная схема есть для кодировки элементов N_2 в виде 1^j0^{2-j}, где $j \in \{1,2\}$, а N_1 — в виде 1^k0^{3-k}, где $k \in \{1,2,3\}$. Причём неважно, как именно упорядочивать элементы всех моноидов. Нужно сначала применить к каждому элементу F_2 существующую по лемме 14 вспомогательную схему, переводящую кодировку 1^i0^{4-i} в кодировки 1^j0^{2-j} и 1^k0^{3-k}, а после всех вычислений — схемы, которые вновь кодируют всё как 1^i0^{4-i}. То же самое рассуждение применим, исследуя F_3 и F_4.

3.3.2. Забывающие ветвящиеся программы ширины 3

F_3 — множество всех функций из $\{1,2,3\}$ в себя. Обозначим $e = \left(\begin{smallmatrix}1&2&3\\1&2&3\end{smallmatrix}\right)$, $N_2 = S_3$, $N_1 = (F_3 \setminus S_3) \cup \{e\}$. Если N_1 — моноид, и $N_1 \setminus \{e\}$ — левый идеал в F_3, то применима лемма 17. Ассоциативность умножения (композиции функций) для N_1 следует из ассоциативности для F_3, единица — это e. Осталось показать замкнутость. Пусть $f_1, f_2 \in N_1$. Если $f_1 = e$, то $f_1 \circ f_2 = f_2 \in N_1$. Иначе $\exists\ i,j \in \{1,2,3\}$: f_1 переводит и i, и j в один и тот же элемент k. Пусть f_2 переводит k в $l \in \{1,2,3\}$. Тогда f_1f_2 переводит и i, и j в l. Значит, $f_1f_2 \in N_1$.

Пусть $f_1 \in F_3$, $f_2 \in N_1$, $f_2 \neq e$. Покажем, что $f_1f_2 \in N_1 \setminus \{e\}$. Если $f_1 \in N_1$, то $f_1f_2 \in N_1$. Причём $\nexists\ f_2^{-1}$, следовательно, $f_1f_2 \neq e$. Пусть $f_1 \notin N_1$. f_2 — не биекция, то есть $\exists\ i,j \in \{1,2,3\}$ такие, что f_2 переводит i и j в один и тот же элемент t. Так как f_1 — биекция, то $\exists\ k,l \in \{1,2,3\}$: $k \neq l$ и f_1 переводит k в i, а l в j. Тогда f_1f_2 переводит и k, и l в t. Значит, $f_1f_2 \in N_1 \setminus \{e\}$.

Осталось построить схему, которая по входу $x_1, x_2, \ldots, x_n \in N_1$ выдаёт их произведение. Если показать, что N_1 — непериодический моноид, то по лемме 18 такая схема существует и принадлежит AC^0. Поскольку все элементы $N_1 \setminus \{e\}$ не биекции, то у них нет обратных элементов. Значит, у N_1 не может быть никаких подгрупп, кроме $\{e\}$. Значит, $3 - OBP \subseteq AC^0$[6].

3.3.3. Забывающие ветвящиеся программы ширины 4

Аналогично рассуждению для F_3, обозначим $N_2 = S_4$, $N_1 = F_4 \setminus S_4 \cup \{e\}$. N_2 и N_1 — моноиды, причём $N_1 \setminus \{e\}$ — левый идеал в F_4 и N_1 непериодический. Как и $4 - PBP$, $4 - OBP \subseteq AC^0$[12].

Заключение.

Булевы схемы — наиболее естественный способ вычисления булевых функций, основанный на том, что любую функцию можно выразить в виде формулы; просто последовательно вычисляем значения формул. AC^0 и ACC^0 — классы булевых схем конечной глубины и полиномиального размера, использующих связки $\vee$, $\wedge$ (AC^0) и MOD (ACC^0). Перестановочные и забывающие ветвящиеся программы — более сложная модель; вычисления сводятся к вычислению композиции перестановок или всюду определённых функций $f : [n] \to [n]$. Чем больше ширина перестановочной/забывающей ветвящейся программы, тем больше класс функций, которые можно вычислить с помощью программы такой ширины. Цель данной работы — дать оценки на эти классы. Результат первой главы — если функция вычислима схемой из NC^1, то она вычислима перестановочной ветвящейся программой ширины 5. Вторая глава состоит из определений и лемм, необходимых для выводов, полученных далее. В третьей главе я получаю оценки вида — если функция вычислима перестановочной (забывающей) программой ширины n (где $n \in \{2, 3, 4\}$), то она вычислима булевой схемой из $AC^0[m]$. На рисунке 3.2 — графическое изображение выводов, полученных в третьей главе.

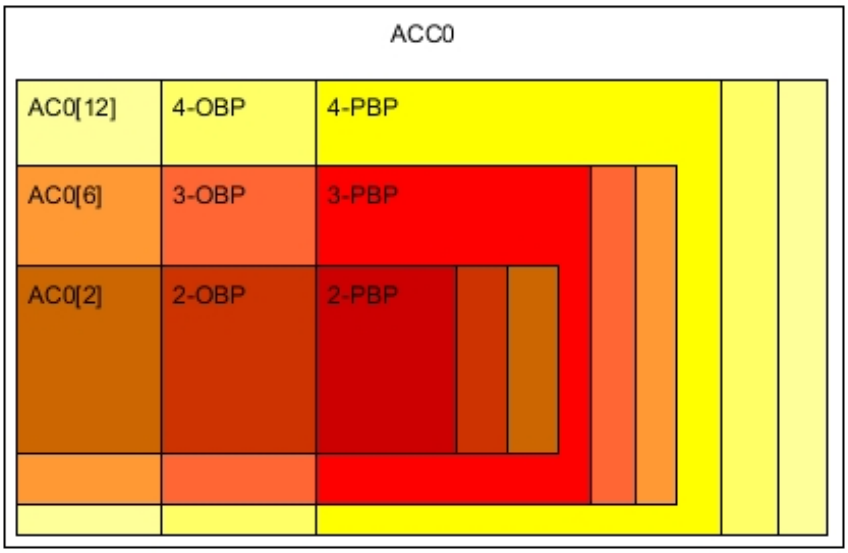

Рис. 3.2: $\forall k : k - PBP \subseteq k - OBP, k - PBP \subseteq (k+1) - PBP, k - OBP \subseteq (k+1) - OBP$. $2 - OBP \subseteq AC^0[2], 3 - OBP \subseteq AC^0[6], 4 - OBP \subseteq AC^0[12]$.

Литература

[1] Stasys Jukna. Boolean Function Complexity: Advances and Frontiers. // Springer, 2012. с. 12-21, 447-451.

[2] Kristoffer Arnsfelt Hansen. An Exposition of the Barrington-Therien Classification. // [Электронный ресурс] services.brics.dk/java/courseadmin/ct12/documents/getDocument/BarringtonTherienExposition.pdf?d=80382.

[3] David Barringhton. Bounded-Width Polynomial-Size Branching Programs Recognize Exactly Those Languages in NC^1. // Journal of Computer and System Sciences 38, 1989. с. 150-16

[4] David Barringhton and Denis Therien. Finite Monoids and the Fine Structure of NC^1. // Journal of the ACM, 1988. с. 941-952.

[5] Joseph Rotman. An Introduction to the Theory of Groups. // Springer, 1995. с. 102-107.

[6] Sanjeev Arora, Boaz Barak. Computational Complexity: A Modern Approach. // Cambridge University Press, 2009. с.106-108.

[7] Claude Shannon. A Symbolic Analysis of Relay and Switching Circuits. // Transactions of American Institute of Electrical Engeneers, 1938.

[8] C. Y. Lee. Representation of Switching Functions by Binary Decision Programs. // Bell System Technical Journal, 1959. с. 985-999.