Курс МФТИ-МИАН «Квантовые вычисления», Весна 2025.

Задачи I: Классические вычисления

Яшин Всеволод Игоревич (yashin.vi@mi-ras.ru)

Здесь собраны задачи в дополнение к лекциям по курсу. На эти задачи полезно посмотреть, или даже решить. Когда в задаче написано "разребитесь в", подразумевается, что желательно выписать доказательство утверждения на бумаге. Решения можно обсудить с лектором очно или прислать оформленные задачи по почте. Работа над задачами примерно следующим образом влияет на итоговую оценку по курсу: если Вы совсем не решали задач, Вы не сможете претендовать на «отлично»; если Вы сдали все задачи, Вы не получите менее, чем «хорошо».

Первая серия задач посвящена изучению общих свойств классических, вероятностных и обратимых булевых схем.

Задача I.1

Докажите универсальность словаря $\{\mathrm{NAND}\}$, где $\mathrm{NAND}(x, y) = \mathrm{NOT} \circ \mathrm{AND}(x, y)$. Иначе говоря, как при помощи бинарной операции NANAD можно представить произвольную булеву функцию?

Задача I.2

Результат Лупанова [1] говорит о том, что любую функцию $f : \{0, 1\}^n \to \{0, 1\}$ можно представить булевой схемой размера $\mathcal{O}(2^n/n)$. Разберитесь и напишите, как работает эта конструкция.

Задача I.3

Булева функция $f : \{0, 1\}^n \to \{0, 1\}^m$ называется афинной, если она имеет вид $f(x) = Ax \oplus b$, где A булева матрица и b некоторый сдвиг. Произвольную афинную булеву функцию можно реализовать при помощи словаря $\{\mathrm{NOT}, \mathrm{XOR}\}$. Покажите, что афинную булеву функцию $f : \{0, 1\}^n \to \{0, 1\}^m$ можно реализовать за $\mathcal{O}(nm)$ вентилей из словаря. Далее, докажите, что произвольную афинную функцию $f : \{0, 1\}^n \to \{0, 1\}^n$ можно реализовать за $\mathcal{O}(\frac{n^2}{\log n})$ операций, и дайте нижнюю оценку.

Задача I.4

Давайте рассмотрим вместо битов систему *дитов* $\{0, \ldots, d - 1\} = \mathbb{Z}_d$, где d некоторое натуральное число. Предложите какой-нибудь словарь операций (конечный, с ограниченными входами), универсальный в классе всех дитных функций, и докажите его универсальность.

Задача I.5

Задача выполнимости на булевых схемах CSAT формулируется следующим образом: на входе подаётся некоторая булева схема C, мы хотим знать, существует ли булевая строка x такая, что $C(x) = 1$. Разберитесь в теореме Кука-Левина о том, что задача CSAT является $\mathtt{NP}$-полной. Тот факт, что CSAT лежит в $\mathtt{NP}$, довольно очевиден: строка x является сертификатом. Но с другой стороны, для доказательства того, что любую $\mathtt{NP}$-задачу можно свести к CSAT, приходится научиться сводить работу машин Тьюринга на данном входе как работу некоторой булевой схемы.

Задача I.6

Класс сложности $\mathtt{NC}^k$ определяется как класс языков, которые разрешаются на словарях с ограниченным входом при помощи равномерных семейств схем с полиномиальным размером и глубиной $\mathcal{O}((\log n)^k)$. Считается, что задачи класса сложности $\mathtt{NC} = \cup_k \mathtt{NC}^k$ поддаются эффективному распараллеливанию, и в этом классе лежит на удивление много важных задач компьютерной алгебры (существует гипотеза, что $\mathtt{P} = \mathtt{NC}$).

Также, зачастую вводят класс языков $\mathtt{AC}^k$ языков, которые разрешаются при помощи равномерных семейств схем с полиномиальным размером и глубиной $\mathcal{O}((\log n)^k)$, но на словарях с *неограниченным* входом. Чем отличаются схемы, используемые в классах $\mathtt{NC}^0$ и $\mathtt{AC}^0$? Покажите, что

$$\mathtt{NC}^k \subseteq \mathtt{AC}^k \subseteq \mathtt{NC}^{k+1}. \tag{1}$$

Отсюда в частности следует, что $\mathtt{NC} = \mathtt{AC} = \cup_k \mathtt{AC}^k$.

Задача I.7

Разберитесь, как можно распараллелить задачу сложения битовых чисел (например см. carry-lookahead adder). Покажите, что эта задача лежит в функциональном аналоге класса $\mathtt{AC}^0 \subseteq \mathtt{NC}^1$ (см. [2, Theorem 1.15]).

Задача I.8

Допустим, для вычислений нам доступны афинные булевы схемы (словарь $\{\mathrm{NOT}, \mathrm{XOR}\}$) и равномерно случайные биты. Покажите, что в таком случае можно реализовать равномерные распределения вероятностей над афинными подпространстваим битовых строк $\mathbb{Z}_2^n$, и только их. Какие распределения вероятности можно реализовать при помощи монотонных схем (словарь $\{\mathrm{AND}, \mathrm{OR}\}$) и равномерно случайных битов?

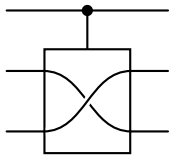
Задача I.9

Докажите, что любое вероятностное отображение можно представить как статистическую смесь детерминированных отображений. Рассмотрим некоторое вероятностное

отображение $\Phi : \ell_1(X) \to \ell_1(X)$, где X конечное множество. Такое отображение называется *бистохастическим*, если оно переводит равномерное распределение на X само в себя. Разберитесь в доказательстве теоремы Биркгофа-фон Неймана: любое бистохастическое отображение есть стастистическая смесь перестановок на X.

Задача I.10

Рассмотрим 3-битный вентиль Фредкина FREDKIN = CSWAP, который удобно обозначать как



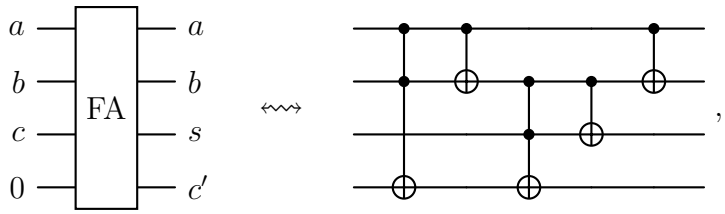
. (2)

Он действует как управляемая замена двух бит: если первый бит в состоянии "1", второй и третий бит меняют значения, иначе ничего не происходит. Выразите вентиль Фредкина через вентили Тоффоли. Покажите, что вентиль Фредкина универсален. Как выразить вентиль Тоффоли при помощи вентилей Фредкина?

Полезно заметить, что у вентиля Фредкина есть свойство *консервативности*: если битовая строка x на входе имеет вес Хэмминга $\mathrm{wt}(x)$ (число единиц в битовой строке), то после действия схемы из вентилей Фредкина вес сохраняется. Такое свойство можно понимать как сохранение числа частиц или энергии в физической системе. Поэтому, если некоторая обратимая операция не сохраняет вес wt, то для её реализации при помощи вентилей Фредкина потребуются анциллы.

Задача I.11

Интересно посмотреть, как необратимо реализовать сложение битовых чисел обратимым образом. Для этого, можно использовать схему вида



, (3)

где $s = a \oplus b \oplus c$ и $c' = \mathrm{maj}(a, b, c)$. Покажите, что эта схема вычисляет то, что надо. Комбинируя такие схемы, можно реализовать сложение битовых чисел за линейное время. Покажите, что можно производить сложение за логарифмическую глубину. Это оказывается важным в некоторых квантовых алгоритмах [3, 4].

Задача I.12

Булева функция f называется *монотонной*, если она сохраняет порядок на битовых строках: из $x \leqslant y$, то $f(x) \leqslant f(y)$. Такие функции выражаются через $\{\mathrm{OR}, \mathrm{AND}\}$.

Покажите, что любая обратимая монотонная булева функция $f : \{0,1\}^n \to \{0,1\}^n$ является некоторой перестановкой бит σ, то есть действует как

$$(x_1, \ldots, x_n) \mapsto (x_{\sigma(1)}, \ldots, x_{\sigma(n)}). \tag{4}$$

Таким образом, в случае обратимых вычислений класс монотонных функций малоинтересен. Оказывается, что в случае обратимых вычислений при возможности свободного использования анцилл единственными невырожденными универсальными классами булевых функций являются класс всех обратимых функций и класс обратимых афинных булевых функций [5].

Задача I.13

Рассмотрим словарь афинных булевых операций $\{\mathrm{NOT}, C\mathrm{NOT}\}$. Пусть задана афинная булева функция $f : \{0,1\}^n \to \{0,1\}^n$, которую можно представить как $f(x) = Ax \oplus c$, где A обратимая булева матрица и c некоторый сдвиг. Убедитесь, что композиция NOT после f изменяет сдвиг c, а действие NOT суммирует одну строку матрицы (A, c) с сдругой строкой. Используя метод Гауссова исключения, разложите функцию f в композицию $\mathcal{O}(n^2)$ элементарных операций $\{\mathrm{NOT}, C\mathrm{NOT}\}$.

Если усовершенствовать Гауссово исключение, то оказывается возможным сделать разложение за $\mathcal{O}(\frac{n^2}{\log n})$ операций [6, 7] – разберитесь в этом алгоритме. Довольно просто также получить нижнюю оценку $\Omega(\frac{n^2}{\log n})$. Посчитайте общее число афинных обратимых операций над $\mathbb{Z}_2$ (см. также [8, Chapter 9]). Пусть b – число схем с одним вентилем. Тогда число функций, собранных из s вентилей, не больше b^s. Получите отсюда нижнюю оценку на s.

---

[1] O. B. Lupanov, On a method of circuit synthesis, Izvestia VUZ **1**, 120 (1958).

[2] H. Vollmer, *Introduction to circuit complexity: a uniform approach* (Springer Science & Business Media, 1999).

[3] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, A logarithmic-depth quantum carry-lookahead adder (2004), arXiv:quant-ph/0406142 [quant-ph].

[4] C. Gidney, Halving the cost of quantum addition, Quantum **2**, 74 (2018).

[5] S. Aaronson, D. Grier, and L. Schaeffer, The classification of reversible bit operations (2015), arXiv:1504.05155 [quant-ph].

[6] K. N. Patel, I. L. Markov, and J. P. Hayes, Efficient synthesis of linear reversible circuits (2003), arXiv:quant-ph/0302002 [quant-ph].

[7] T. G. De Brugière, M. Baboulin, B. Valiron, S. Martiel, and C. Allouche, Gaussian elimination versus greedy methods for the synthesis of linear reversible circuits, ACM Transactions on Quantum Computing **2**, 1–26 (2021).

[8] P. J. Cameron, *Combinatorics: topics, techniques, algorithms* (Cambridge University Press, 1994).