

1. Коммуникационная сложность

Алиса и Боб играют в следующую игру. Алисе сообщается число x , а Бобу — число y , также им задана функция f от двух аргументов, и они хотят вычислить её значение $f(x, y)$ на некотором входе. В их распоряжении есть устройство связи, которое позволяет передавать друг другу битовые сообщения (т.е. за одно сообщение можно послать «0» или «1»). Алиса и Боб могут заранее договориться о том, какие сообщения они будут посылать.

Определение 1 (Коммуникационный протокол)

Пусть X , Y и Z — это три произвольных непустых конечных множества, а $f: X \times Y \rightarrow Z$ — некоторая функция. *Коммуникационный протокол* Π для функции f — это упорядоченное корневое двоичное дерево со следующими пометками:

- каждая внутренняя вершина помечена буквой «А» или «Б»,
- каждое ребро к левому потомку помечено нулём, к правому — единицей,
- каждый лист помечен элементом множества Z .

Для каждой внутренней вершины v с пометкой «А» определена функция $A_v: X \rightarrow \{0, 1\}$, а для каждой внутренней вершины u с пометкой «Б» определена функция $B_u: Y \rightarrow \{0, 1\}$.

Результат протокола Π на входе (x, y) обозначается $\Pi(x, y)$, и определяется, как пометка конечной вершины пути $\pi(x, y)$, построенного по следующим правилам:

- первая вершина пути $\pi(x, y)$ — это корень,
- каждая следующая вершина пути является потомком предыдущей, причём
 - каждая вершина пути v с пометкой «А» соединена с потомком ребром с пометкой $A_v(x)$
 - каждая вершина u с пометкой «Б» соединена с потомком ребром с пометкой $B_u(y)$
- последняя вершина пути $\pi(x, y)$ — лист.

Протокол Π называется *корректным* протоколом для функции f , если для каждой пары входов (x, y) выполняется $\Pi(x, y) = f(x, y)$.

Протокол описывает общение игроков на всех возможных входах. Пометки во внутренних вершинах — это указание на игрока, который посылает сообщение, пометки на рёбрах — это посылаемые сообщения, пометки в листьях — это результат вычисления $f(x, y)$, а функции в вершинах — это правила, по которым игроки выбирают сообщение, которое нужно послать в данный момент. Каждой паре входов (x, y) соответствует путь $\pi(x, y)$ от корня к некоторому листу, который задаётся описанными выше правилами. Если протокол корректный, то для каждой пары входов (x, y) путь $\pi(x, y)$ заканчивается в листе с пометкой $f(x, y)$.

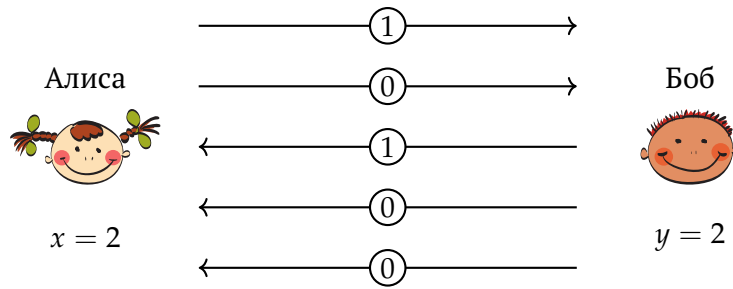


Рис. 1. Возможное взаимодействие Алисы и Боба.

Определение 2

Сложностью функции f называется наименьшая глубина протокола, вычисляющего функцию f (обозначается $C(f)$). (Глубина дерева — это максимальная рёберная длина пути от корня до листа.)

Также обозначим за $L(f)$ — минимальное число листьев в протоколе для функции f .

Входное пространство коммуникационной задачи можно воспринимать как матрицу. Каждой функции f будем сопоставлять матрицу $X \times Y$ (M_f), в которой в клетке (x_i, y_j) стоит значение $f(x_i, y_j)$.

Утверждение 1

Рассмотрим дерево протокола со входом из множества $X \times Y$. Рассмотрим в нём произвольную вершину u . Тогда все входы, из которых можно прийти в вершину u , образуют комбинаторный прямоугольник $R_u = X_u \times Y_u \subseteq X \times Y$.

Доказательство. Это можно доказать двумя способами.

Первый способ: пусть на входах (x_1, y_1) и (x_2, y_2) мы приходим в вершину u . Тогда нетрудно убедиться, что на входе (x_1, y_2) Алиса и Боб будут делать те же действия, что и на входах (x_1, y_1) и (x_2, y_2) соответственно. Отсюда видно, что входы, приводящие в вершину u , образуют прямоугольник $R_u = X_u \times Y_u \subseteq X \times Y$.

Второй способ: Рассмотрим таблицу элементов $X \times Y$. После первого хода Боба табличка делится пополам горизонтальной линией, так как при одних $x \in X$ Боб посылает Алисе 1, а при других — 0. Потом Алиса посылает свой бит Бобу, и каждый из двух получившихся прямоугольников делится своей вертикальной прямой, и так далее. В итоге мы получим разбиение $X \times Y$ на непересекающиеся прямоугольники, и каждый из этих прямоугольников соответствует листу в коммуникационном протоколе. \square

Про прямоугольник R_u можно думать в следующем образом: если мы находимся в вершине протокола u , то нам необходимо решить задачу (то есть построить протокол) для всех входов из прямоугольника R_u . В частности этот подход можно рассмотреть, как комбинаторное определение протокола: бинарное дерево, в котором каждой вершине сопоставлен прямоугольник входов. И если вершины a, b являются потомками u , то $R_u \subseteq R_a \cup R_b$.

Определение 3

Прямоугольник $R \subset X \times Y$ называется одноцветным для функции f , если существует $z \in Z$, что для всех $(x, y) \in R$ верно $f(x, y) = z$. Такой прямоугольник будем называть z -одноцветным.

Заметим, что в листьях коммуникационного протокола находятся одноцветные прямоугольники.

Допустим, что игрокам нужно вычислить функция $f: \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$. Рассмотрим величину $\chi_0(f)$, равную минимальному числу комбинаторных (то есть это не обязательно геометрически один прямоугольник) прямоугольников, которыми можно дизъюнктно покрыть нули в таблице. Аналогично определяется $\chi_1(f)$. Тогда листьев в коммуникационном протоколе будет хотя бы $\chi_0(f) + \chi_1(f)$. Эти рассуждения дают следующую оценку:

$$C(f) \geq \log(\chi_0(f) + \chi_1(f)).$$

Однако эта оценка не всегда точна. Теперь рассмотрим пример, когда разбиение на одноцветные прямоугольники может не соответствовать протоколу.

Пример 1. Рассмотрим такой пример разбиения таблицы $X \times Y$ на прямоугольники: в центре находится прямоугольник из 1, а вокруг него расположены 4 прямоугольника из 0. Покажем, что для этого разбиения не существует дерева протокола. Действительно, рассмотрим первое действие игроков. После него таблица должна поделиться на две части, но на рисунке 2 видно, что нет разреза, проходящего через всю таблицу.

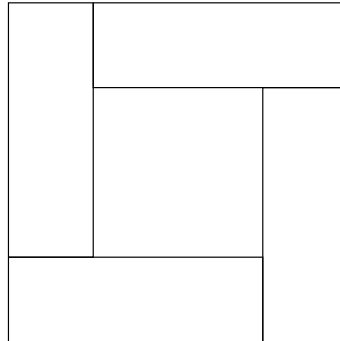


Рис. 2.

2. Задачи

Р 1. Докажите, что $C(EQ_n) \geq n$, где $EQ_n: \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ и $EQ_n(x, y) = 1 \Leftrightarrow x = y$.

Р 2. Рассмотрим функцию $SUM: \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^{n+1}$, возвращающую сумму чисел в двоичной системе счисления. Докажите, что $C(SUM) = 2n$.

Р 3. Пусть дан граф G без петель. Алиса и Боб получают две вершины данного графа x, y и хотят узнать существует ли ребро (x, y) . Докажите, что детерминированная сложность данной задачи не менее $\log \chi(G)$, где $\chi(G)$ — хроматическое число графа G .

Подсказка: попробуйте предъявить хорошую раскраску, если есть короткий коммуникационный протокол.

Р 4. Покажите, что $C(\text{MED}) = \mathcal{O}(\log^2 n)$, где x и y это характеристические функции подмножеств $[n]$, а $\text{MED}(x, y)$ — медиана мультимножества $x \cup y$ (если элемент встречается и в x и в y , то считаем его дважды).

Комментарий: на самом деле $C(\text{MED}) = \Theta(\log n)$.

Р 5. У Алисы имеется n -битная строка x , а у Боба n -битная строка y . Известно, что y получен из x инвертированием одного бита.

- Придумайте детерминированный коммуникационный протокол сложности $\mathcal{O}(\log n)$, который позволяет Бобу узнать x .
- Придумайте однораундовый детерминированный коммуникационный протокол сложности $\mathcal{O}(\log n)$, который позволяет Бобу узнать x . (В однораундовом протоколе Алиса посылает некоторое сообщение Бобу, после чего Боб вычисляет результат).

Р 4.2. *Обобщенное неравенство Фано.* Пусть случайная величина α принимает значения в некотором n элементном множестве A . Пусть значение случайной величины β принадлежит A с вероятностью p , причём условная вероятность события $\alpha \neq \beta$ при условии $\beta \in A$ равна ε . Докажите, что выполняется неравенство:

$$H(\alpha \mid \beta) \leq (1 - p) \log n + p\varepsilon \log(n - 1) + ph(\varepsilon).$$

Р 3.4. Пусть $G = (V, E)$ неориентированный граф, t — число треугольников и ℓ — число ребер. Докажите, что $(6t)^2 \leq (2\ell)^3$.

Р 1.7. Даны две группы камешков, причем камешки в каждой группе упорядочены по весу. В первой группе n камешков, а во второй — m . Требуется упорядочить все камешки по весу. Придумайте способ, решающий эту задачу за менее чем $m \log n$ взвешиваний при больших n .

