# The Tiling Method

Max von Hippel

March 24, 2021

## 1 Problem Statement

Consider a two-party communication problem, in which the participants

(a) Alice

and

(b) Bob

*participate* to compute a function:

$$f : \underbrace{\mathbb{B}^n}_{\substack{\text{Alice's}\\\text{input}}} \times \underbrace{\mathbb{B}^n}_{\substack{\text{Bob's}\\\text{input}}} \to \underbrace{\mathbb{B}}_{\substack{\text{global}\\\text{output}}}$$

The players can come up with a *protocol* $\Pi = (p_1, \ldots, p_t)$, namely, for some natural $t \in \mathbb{N}$, a sequence of $t$-many functions $p_i : \mathbb{B}^* \to \mathbb{B}^*$ such that the communication between the players looks like this:

Alice is given input $x$.

Hi Bob. I'm not divulging $x$, but, $p_1(x) = \texttt{P1}$.

Bob is given input $y$.

Thanks Alice. I'm not divulging $y$, but, $p_2(y, \texttt{P1}) = \texttt{P2}$.

Thanks Bob. Don't tell anyone, but: $p_3(x, \texttt{P1}, \texttt{P2}) = \texttt{P3}$.

Is that so? Well, $p_4(y, \texttt{P1}, \texttt{P2}, \texttt{P3}) = \texttt{P4}$.

Wicked. In that case, $p_5(x, \texttt{P1}, \texttt{P2}, \texttt{P3}, \texttt{P4}) = \texttt{P5}$.

*...yada yada yada...*

$p_t(x, \texttt{P1}, \texttt{P2}, \texttt{P3}, \texttt{P4}, \ldots, \texttt{P(t-1)}) = \texttt{Pt}$, and TTFN!

Critically, the functions $p_i$ can be *anything* so long as they are well-defined. For example, $p_2$ could be the function that asks if $\langle y, \texttt{P2} \rangle$ is a word in $A_{TM}$.

**Definition 1** (Communication Complexity)**.** *Suppose* $\Pi$ *is a protocol for* $f$ *in which at most* $t$ *bits are communicated between Alice and Bob. Then the* communication complexity *of* $\Pi$*, denoted* $C(\Pi)$*, is* $t$*. The communication complexity of* $f$*, denoted* $C(f)$*, is the minimum communication complexity achieved by any protocol for* $f$*.*

Given some such function $f$, it would be nice if we could automatically compute a reasonable lower bound on its communication complexity.

## 2 The Tiling Method

One way to do this is with the *tiling method*. We will give the method immediately, and in tandem, we will illustrate the method using the function $f(x,y) = x < y$ where $x,y$ are integers in $\{0,1,2,3\}$, encoded in $\mathbb{B}$oolean.

**Definition 2** ($M(f)$)**.** *The* matrix of $f$*, denoted* $M(f)$*, is the* $2^n \times 2^n$ *matrix whose* $(x,y)$*th entry is the value* $f(x,y)$*.*

**Definition 3** (Combinatorial Rectangle)**.** *A* combinatorial rectangle *in* $M(f)$ *is any submatrix of* $M$*. We say a rectangle* $A \times B$ *in* $M(f)$ *is* monochromatic *if for all* $x,x'$ *in* $A$ *and* $y,y'$ *in* $B$*,* $M_{x,y} = M_{x',y'}$*.*

Each message-send event in a protocol $\Pi$ splits $M(f)$ into two or more combinatorial rectangles of still-possible values for $f(x,y)$. An example is given below, using the LeastSignficantBit protocol for <, with Alice sending the first message.

| | 000 | 001 | 010 | 011 | 100 |
|---|---|---|---|---|---|
| 000 | 0 | 1 | 1 | 1 | 1 |
| 001 | 0 | 0 | 1 | 1 | 1 |
| 010 | 0 | 0 | 0 | 1 | 1 |
| 011 | 0 | 0 | 0 | 0 | 1 |
| 100 | 0 | 0 | 0 | 0 | 0 |

Alice: "$x =$ __0"

| | 000 | 001 | 010 | 011 | 100 |
|---|---|---|---|---|---|
| 000 | 0 | 1 | 1 | 1 | 1 |
| 010 | 0 | 0 | 0 | 1 | 1 |
| 100 | 0 | 0 | 0 | 0 | 0 |

Alice: "$x =$ __1"

| | 000 | 001 | 010 | 011 | 100 |
|---|---|---|---|---|---|
| 001 | 0 | 0 | 1 | 1 | 1 |
| 011 | 0 | 0 | 0 | 0 | 1 |

Figure 2: The matrix $M(<)$ for inputs $x,y \in \{0,1,2,3\}$. Values of $x$ are given in the rows, while values of $y$ are given in the columns. False (i.e. 0) values are marked red for clarity. We show how in the LeastSignificantBit protocol, $M(<)$ can be partitioned into two rectangles depending on the substance of the initial message sent by Alice.

Since every protocol has finite length, a run of a protocol can only split $M(f)$ finitely many times. Hence each protocol $\Pi$ of $f$ induces a tree of combinatorial rectangles, rooted at $M(f)$, where the leaves represent the matrix of possible values of $f(x,y)$ once the protocol has concluded. By definition, a protocol must conclude with both participants knowing the value $f(x,y)$. Therefore the leaves of each such tree must be monochromatic.

Now we get to the punchline.

**Definition 4** (Monochromatic Tiling)**.** *A* monochromatic tiling *of $M(f)$ is a partition of $M(f)$ into disjoint monochromatic rectangles.*

**It's thinking time.**

*Notice* that if $\Pi$ is a protocol for $f$, then the leaves of the tree induced by $\Pi$ and rooted at $M(f)$ clearly form a monochromatic tiling of $M(f)$.

*Recall* from eons ago, when you were an undergrad and had to know useful things, that the number of leaves in a binary tree can be used to upper-bound its depth.

*Realize* that the depth of the binary tree induced by $\Pi$ is exactly $C(\Pi)$.

*Observe* that although we made the math easy by assuming bit-sized messages, this idea clearly generalizes.

Let $\chi(f)$ denote the minimum number of rectangles in any monochromatic tiling of $M(f)$.

**Theorem 1** (The Punchline)**.**

$$log_2\chi(f) \leq C(f) \leq 16\big(log_2\chi(f)\big)^2$$

*Proof.* We need to show the following.

(a) $\log_2\chi(f) \leq C(f)$

(b) and, $C(f) \leq 16\big(\log_2\chi(f)\big)^2$.

We prove (a) then (b). Suppose that $f$ has communication complexity $C(f)$. Then there exists a protocol $\Pi$ in which at most $C(f)$ bits are communicated between the two participants. For simplicity suppose each bit communicated is an individual message. Then $\Pi$ induces a tree whose maximum depth is $C(f)$, and which induces a monochromatic partition of $M(f)$. Since every monochromatic partition of $M(f)$ requires at least $\chi(f)$ rectangles, clearly the tree induced by $\Pi$ must have at least $\chi(f)$ leaves. But we're talking about a binary tree so it immediately follows that the tree must have at least $\log_2\chi(f)$ depth. So (a) holds and we are done.

Now let's prove (b). Consider the function $f : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$. There are, by definition, at most $\chi(f) \geq 1$ distinct values $f(x,y)$.

If $\chi(f) = 1$, then there is only one possible value of $f(x,y)$, and so $C(f) = 0$. In this case we get $C(f) = 0 \leq 0 = 16\log_2^2 1 = 16\log_2^2\chi(f)$, so (b) holds and we are done.

If $\chi(f) = 2$, then there are 2 possible values of $f(x,y)$, say, $\alpha$ and $\beta$. We can partition $\mathbb{B}^n$ into the spaces $X, Y$ where if $x \in X$ then $f(x,y) = \alpha$ if and only if $y \in Y$. Then Alice could send a single bit indicating if $x \in X$, and Bob could reply with a single bit indicating if $y \in Y$, at which point both Alice and Bob would immediately know the value of $f(x,y)$. This protocol communication complexity 2, as only 2 bits need to be communicated. We conclude that if $\chi(f) = 2$, then $2 = C(f) \le 16\left(\log_2 \chi(f)\right)^2 = 16\left(\log_2 2\right)^2 = 16(1)^2 = 16$, so (b) holds and we are done.

For the inductive step, suppose that whenever $\chi(f) = k$ for some integer $k \ge 2$, it immediately follows that $C(f) \le 16\left(\log_2 \chi(f)\right)^2$. We want to show the same holds for $k+1$. Suppose $\chi(f) = k+1$. Choose some possible value $z = f(x,y)$ arbitrarily from the $\le k+1$ options. Partition the space $\mathbb{B}^n$ into sets $X, y$ such that whenever $x \in X$, $f(x,y) = z$ if and only if $y \in Y$. Have Alice send a single bit $b$ to begin the protocol, indicating if $x \in X$, and have Bob reply with a bit $b'$ indicating if $y \in Y$. At that point if $f(x,y) = z$ the protocol is over, otherwise it reduces to a protocol over $k$ possible colors and, by our inductive assumption, (b) holds.

□

Notice that the tiling method directly relates to the fooling set method. Specifically, if $f$ has a fooling set with $m$ pairs, then $\chi(f) \ge m$.

*Proof.* Suppose $(x,y), (\bar{x},\bar{y})$ are two of the pairs in the fooling set. Then they cannot be in a monochromatic rectangle, since by definition, the following set contains at least two distinct values:

$$\left\{f(x,y), f(x,\bar{y}), f(\bar{x},y) f(\bar{x},\bar{y})\right\}$$

□