1 Linear Programs, Polytopes, and Extended Formulations

1.1 Linear Programs for Combinatorial Optimization Problems

You’ve probably seen some polynomial-time algorithms for the problem of computing a maximum-weight matching of a bipartite graph. Many of these, like the Kuhn-Tucker algorithm, are “combinatorial algorithms,” meaning that all of its steps work directly with the graph.

Linear programming is also an effective tool for solving many discrete optimization problems. For example, consider the following linear programming relaxation of the maximum-weight bipartite matching problem (for a weighted bipartite graph $G = (U, V, E, w)$):

$$\max \sum_{e \in E} w_e x_e$$  \hspace{1cm} (1)

subject to

$$\sum_{e \in \delta(v)} x_e \leq 1$$ \hspace{1cm} (2)

for every vertex $v \in U \cup V$ (where $\delta(v)$ denotes the edge incident to $v$) and

$$x_e \geq 0$$ \hspace{1cm} (3)
for every edge $e \in E$. In this formulation, each decision variable $x_e$ is intended to encode whether an edge $e$ is in the matching ($x_e = 1$) or not ($x_e = 0$). It is an easy exercise to verify that the vectors of $\{0, 1\}^E$ that satisfy the constraints (2) and (3) are precisely the characteristic vectors of the matchings of $G$, with the objective function value of the solution to the linear program equally the total weight of the matching.

Since every characteristic vector of a matching satisfies (2) and (3), and the set of feasible solutions to the linear system defined by (2) and (3) is convex, the convex hull of the characteristic vectors of matchings is contained in this feasible region.\(^2\) Also note that every characteristic vector $\mathbf{x}$ of a matching is a vertex\(^3\) of this feasible region — since all feasible solutions have all coordinates bounded by 0 and 1, the 0-1 vector $\mathbf{x}$ cannot be written as a non-trivial convex combination of other feasible solutions. The worry is, does this feasible region contain anything other than the convex hull of characteristic vectors of matchings? Equivalently, does it have any vertices that are fractional, and hence do not correspond to matchings? (Note that integrality is not explicitly enforced by (2) or (3).)

A nice fact is that the vertices of the feasible region defined by (2) and (3) are precisely the characteristic vectors of matchings of $G$. This is equivalent to the Birkhoff-von Neumann theorem (see exercises). There are algorithms that solve linear programs in polynomial time (and output a vertex of the feasible region, see e.g. \([?]\)), so this implies that the maximum-weight bipartite matching problem can be solved efficiently using linear programming.

How about the more general problem of maximum-weight matching in general (non-bipartite) graphs? While the same linear system (2) and (3) still contains the convex hull of all characteristic vectors of matchings, and these characteristic vectors are vertices of the feasible region, there are also other, fractional, vertices. To see this, consider the simplest non-bipartite graph, a triangle. Every matching contains at most 1 edge. But assigning each edge an $x$-value of $\frac{1}{2}$ yields a fractional solution that satisfies (2) and (3). This solution clearly cannot be written as a convex combination of characteristic vectors of matchings.

It is possible to add to (2) additional inequalities — “odd cycle inequalities” stating that, for every odd cycle $C$ of $G$, $\sum_{e \in C} x_e \leq (|C| - 1)/2$ — to recover the property that feasible solutions are precisely the convex combinations of characteristic vectors of matchings? Unfortunately, many graphs have an exponential number of odd cycles. Is it possible to add only polynomially inequalities instead? Unfortunately not — the convex hull of the characteristic vectors of matchings can have $2^{\Omega(n)}$ “facets” \([?]\).\(^4\) We define facets more for-

---

\(^2\)Recall that a set $S \subseteq \mathbb{R}^n$ is convex if it is “filled in,” with $\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \in S$ whenever $\mathbf{x}, \mathbf{y} \in S$ and $\lambda \in [0, 1]$. Recall that the convex hull of a point set $P \subseteq \mathbb{R}^n$ is the smallest (i.e., intersection of all) convex set that contains it. Equivalently, it is the set of all finite convex combinations of points of $P$, where a convex combination has the form $\sum_{i=1}^p \lambda_i \mathbf{x}_i$ for non-negative $\lambda_i$’s summing to 1 and $\mathbf{x}_1, \ldots, \mathbf{x}_p \in P$.

\(^3\)There is an unfortunate clash of terminology when talking about linear programming relaxations of combinatorial optimization problems: a “vertex” might refer to a node of a graph or to a “corner” of a geometric set.

\(^4\)This linear programming formulation does still lead to a polynomial-time algorithm, but using fairly heavy machinery — the “ellipsoid method” \([?]\) and a “separation oracle” for the odd cycle inequalities \([?]\). There are also polynomial-time combinatorial algorithms for non-bipartite matching, beginning with Edmonds \([?]\).
mally in Section ??, but intuitively they are the “sides” of a polytope,\(^5\) like the \(2n\) sides of an \(n\)-dimensional cube. It is intuitively clear that a polytope with \(\ell\) facets needs \(\ell\) inequalities to describe — it’s like cleaving a shape out of marble, with each inequality contributing a single cut. We conclude that there is no linear program of polynomial size that captures the maximum-weight bipartite matching problem.

### 1.2 Auxiliary Variables and Extended Formulations

Our exponential lower bound on the number of linear inequalities needed to describe the convex hull of characteristic vectors of matchings of a non-bipartite graph applies to linear systems in \(\mathbb{R}^E\), with one dimension per edge. The idea of an *extended formulation* is to add a polynomial number of auxiliary decision variables, with the hope that radically fewer inequalities are needed to describe the region of interest in the higher-dimensional space.

This idea might sound like grasping at straws, but sometimes it actually works. For example, fix a positive integer \(n\), and represent a permutation \(\pi \in S_n\) by the \(n\)-vector \(x_\pi = (\pi(1), \pi(2), \ldots, \pi(n))\), with all coordinates in \(\{1, 2, \ldots, n\}\). The *permutahedron* is the convex hull of all \(n!\) such vectors. The permutahedron is known to have \(2^n/2\) facets [?], so a polynomial-sized linear description would seem to be out of reach.

Suppose we add \(n^2\) auxiliary variables, \(y_{ij}\) for all \(i, j \in \{1, 2, \ldots, n\}\). The intent is for \(y_{ij}\) to be a 0-1 variable that indicates whether or not \(\pi(i) = j\) — in this case, the \(y_{ij}\)’s simply form the \(n \times n\) permutation matrix that corresponds to \(\pi\).

We next add a set of constraints to enforce the desired semantics of the \(y_{ij}\)’s (cf., (2) and (3)):

\[
\sum_{j=1}^{n} y_{ij} \leq 1 \tag{4}
\]

for \(i = 1, 2, \ldots, n\),

\[
\sum_{i=1}^{n} y_{ij} \leq 1 \tag{5}
\]

for \(j = 1, 2, \ldots, n\), and

\[
y_{ij} \geq 0 \tag{6}
\]

for all \(i, j \in \{1, 2, \ldots, n\}\). We also add constraints enforcing consistency between the permutation encoded by the \(x_i\)’s and by the \(y_{ij}\)’s:

\[
x_i = \sum_{i=1}^{n} j y_{ij} \tag{7}
\]

for all \(i = 1, 2, \ldots, n\).

\(^5\)A *polytope* is just a high-dimensional polygon — an intersection of halfspaces that is bounded or, equivalently, the convex hull of a finite set of points.
It is straightforward to check that the vectors $y \in \{0, 1\}^{n^2}$ that satisfy (4)–(6) are precisely the permutation matrices. For such a $y$ corresponding to a permutation $\pi$, the constraints (7) forces the $x_i$'s to encode the same permutation $\pi$. Using again the Birkhoff-von Neumann Theorem, every vector $y \in \mathbb{R}^{n^2}$ that satisfies (4)–(6) is a convex combination of permutation matrices (see exercises). Constraint (7) implies that the $x_i$'s encode the same convex combination of permutations. Thus, if we take the set of solutions in $\mathbb{R}^{n \times n^2}$ that satisfy (4)–(7) and project onto the $x$-coordinates, we get exactly the permutahedron. This is what we mean by an extended formulation of a polytope, in this case the permutahedron.

To recap the remarkable trick we just pulled off: blowing up the number of variables from $n$ to $n + n^2$ reduced the number of inequalities needed from $2^{n/2}$ to $n^2 + 3n$. This allows us to optimize a linear function over the permutahedron in polynomial time. Given a linear function (in the $x_i$'s), we optimize it over the (polynomial-size) extended formulation, and retain only the $x$-variables of the optimal solution.

Given the utility of polynomial-size extended formulations, we’d obviously like to understand which problems have them. For example, does the non-bipartite matching problem admit such a formulation? The goal of these lectures is to develop communication complexity-based techniques for ruling out such polynomial-size extended formulations. We’ll prove an impossibility result for the “correlation polytope;” similar (but much more involved) arguments imply that every extended formulation of the non-bipartite matching problem requires an exponential number of inequalities [?].

**Remark 1.1 (Geometric Intuition)** It may seem surprising that adding a relatively small number of auxiliary variables can radically reduce the number of inequalities to describe a set — equivalently, that projecting onto a subset of variables can massively blowup the number of sides. It’s hard to draw (low-dimensional) pictures that illustrate this point. Playing around with projections of three-dimensional polytopes onto the plane, one can observe that non-facets of the high-dimensional polytope (edges) often become facets (again, edges) in the low-dimensional projection. Since the number of lower-dimensional faces of a polytope can be much bigger than the number of facets — already in the 3-D cube, there are 12 edges and only 6 sides — it should be plausible that a projection could significantly increase the number of facets.

## 2 Nondeterministic Communication Complexity

The connection between extended formulations of polytopes and communication complexity involves nondeterministic communication complexity. We studied this model implicitly in parts of Lecture #4; this section makes the model explicit.

Consider a function $f : X \times Y \rightarrow \{0, 1\}$ and the corresponding 0-1 matrix $M(f)$, with rows indexed by Alice’s possible inputs and columns indexed by Bob’s possible inputs. In Lecture #4 we proved that if every covering of $M(f)$ by monochromatic rectangles\(^6\) requires

---

\(^6\)Recall that a rectangle is a subset $S \subseteq X \times Y$ that has a product structure, meaning $S = A \times B$ for some $A \subseteq X$ and $B \subseteq Y$. Equivalently, $S$ is closed under “mix and match:” whenever $(x_1, y_1)$ and $(x_2, y_2)$
at least \( t \) rectangles, then the deterministic communication complexity of \( f \) is at least \( \log_2 t \).

The reason is that every communication protocol computing \( f \) with communication cost \( c \) induces a partition of \( M(f) \) into at most \( 2^c \) monochromatic rectangles, and partitions are a special case of coverings. See also Figure ??.

Communication complexity lower bounds that are proved through coverings are actually much stronger than we’ve let on thus far — they apply also to nondeterministic protocols, which we define next.

You presumably already have a feel for nondeterminism from your study of the complexity class \( NP \). Recall that one way to define membership in \( NP \) is via the existence of a polynomial-length and polynomial-time verifiable certificate of membership. To see how an analog might work with communication protocols, consider the complement of the Equality problem, \( \neg \text{Equality} \). If a third party wanted to convince Alice and Bob that their inputs \( x \) and \( y \) are different, it would not be difficult: just specify an index \( i \in \{1, 2, \ldots, n\} \) for which \( x_i \neq y_i \). Specifying an index requires \( \log_2 n \) bits, and specifying whether or not \( x_i = 0 \) and \( y_i = 1 \) or \( x_i = 1 \) and \( y_i = 0 \) requires one additional bit. Given such a specification, Alice and Bob can check the correctness of this “proof of non-equality” without any communication. If \( x \neq y \), there is always a \( (\log_2 +1) \)-bit proof that will convince Alice and Bob of this fact; if \( x = y \), then no such proof will convince Alice and Bob otherwise. This means that \( \neg \text{Equality} \) has nondeterministic communication complexity at most \( \log_2 n + 1 \).

Coverings of \( M(f) \) by monochromatic rectangles are closely related to the nondeterministic communication complexity of \( f \). We first show how coverings lead to nondeterministic protocols. It’s easiest to formally define such protocols after the proof.

**Proposition 2.1** Let \( f : X \times Y \to \{0, 1\} \) be a Boolean function and \( M(f) \) the corresponding matrix. If there is a cover of the 1-entries of \( M(f) \) by \( t \) 1-rectangles, then there is a nondeterministic protocol that verifies \( f(x, y) = 1 \) with cost \( \log_2 t \).

**Proof:** Let \( R_1, \ldots, R_t \) denote a covering of the 1s of \( M(f) \) by 1-rectangles. Alice and Bob can agree to this covering in advance of receiving their inputs. Now consider the following scenario:

1. A prover — a third party — sees both inputs \( x \) and \( y \). (This is the formal model used for nondeterministic protocols.)

2. The prover writes an index \( i \in \{1, 2, \ldots, t\} \) — the name of a rectangle \( R_i \) — on a blackboard, in public view. Since \( R_i \) is a rectangle, it can be written as \( R_i = A_i \times B_i \) with \( A_i \subseteq X, B_i \subseteq Y \).

3. Alice accepts if and only if \( x \in A_i \).

4. Bob accepts if and only if \( y \in B_i \).

This protocol has the following properties:

---

are in \( S \), so is \( (x_1, y_2) \) and \( (x_2, y_1) \). A rectangle is monochromatic (w.r.t. \( f \)) if it contains only 1-entries of \( M(f) \) or only 0-entries of \( M(f) \). In these cases, we call it a 1-rectangle or a 0-rectangle, respectively.
1. If \( f(x, y) = 1 \), then there exists a proof such that Alice and Bob both accept. (Since \( f(x, y) = 1 \), \((x, y) \in R_i\) for some \(i\), and Alice and Bob both accept if “\(i\)” is written on the blackboard.)

2. If \( f(x, y) = 0 \), there is no proof that both Alice and Bob accept. (Whatever index \( i \in \{1, 2, \ldots, t\} \) is written on the blackboard, since \( f(x, y) = 0 \), either \( x \not\in R_i \) or \( y \not\in R_i \), causing a rejection.)

3. The maximum length of a proof is \( \log_2 t \). (A proof is just an index \( i \in \{1, 2, \ldots, t\} \).)

These three properties imply, by definition, that the nondeterministic communication complexity of the function \( f \) and the output 1 is at most \( \log_2 t \). ■

The proof of Proposition 2.1 introduces the formal model of nondeterministic communication complexity: Alice and Bob are given a “proof” or “advice string” by a prover, which can depend on both of their inputs; the communication cost is the worst-case length of the proof; and a protocol is said to compute an output \( z \in \{0, 1\} \) of a function \( f \) if \( f(x, y) = z \) if and only if there exists proof such that both Alice and Bob accept.

With nondeterministic communication complexity, we speak about both a function \( f \) and an output \( z \in \{0, 1\} \). For example, if \( f \) is \text{EQUALITY}, then we saw that the nondeterministic communication complexity of \( f \) and the output 0 is at most \( \log_2 n + 1 \). Since it’s not clear how to convince Alice and Bob that their inputs are equal without specifying at least one bit for each of the \( n \) coordinates, one might expect the nondeterministic communication complexity of \( f \) and the output 1 to be roughly \( n \). (And it is, as we’ll see.)

The way we’ve defined nondeterministic protocols, Alice and Bob never speak, they only verify. This is without loss of generality, since given a protocol in which they do speak, one could modify it so that the prover writes on the blackboard everything that they would have said. We encourage the reader to formalize an alternative definition of nondeterministic protocols without a prover and in which Alice and Bob speak nondeterministically, and to prove that this definition is equivalent to the one we’ve given above (see Exercises).

Next we prove the converse of Proposition 2.2.

**Proposition 2.2** If the nondeterministic communication complexity of the function \( f \) and the output 1 is \( c \), then there is a covering of the 1s of \( M(f) \) by \( 2^c \) 1-rectangles.

**Proof:** Let \( \mathcal{P} \) denote a nondeterministic communication protocol for \( f \) and the output 1 with communication cost (i.e., maximum proof length) at most \( c \). For a proof \( \ell \), let \( Z(\ell) \) denote the inputs \((x, y)\) where both Alice and Bob accept the proof. We can write \( Z(\ell) = A \times B \), where \( A \) is the set of inputs \( x \in X \) of Alice where she accepts the proof \( \ell \), and \( B \) is the set of inputs \( y \in Y \) of Bob where he accepts the proof. By the assumed correctness of \( \mathcal{P} \), \( f(x, y) = 1 \) for every \((x, y) \in Z(\ell)\). That is, \( Z(\ell) \) is a 1-rectangle.

By the first property of nondeterministic protocols, for every 1-input \((x, y)\) there is a proof such that both Alice and Bob accept. That is, \( \bigcup_{\ell} Z(\ell) \) is precisely the set of 1-inputs of \( f \) — a covering of the 1s of \( M(f) \) by 1-rectangles. Since the communication cost of \( \mathcal{P} \) is at most \( c \), there are at most \( 2^c \) different proofs \( \ell \). ■
Proposition 2.2 implies that communication complexity lower bounds derived from covering lower bounds apply to nondeterministic protocols.

**Corollary 2.3** If every covering of the 1s of $M(f)$ by 1-rectangles uses at least $t$ rectangles, then the nondeterministic communication complexity of $f$ is at least $\log_2 t$.

Thus our arguments in Lecture #4, while simple, were even more powerful than we realized — they prove that the nondeterministic communication complexity of EQUALITY, DISJOINTNESS, and GREATER-THAN (all with output 1) is at least $n$. It’s kind of amazing that these lower bounds can be proved with so little work.

### 3 Extended Formulations and Nondeterministic Communication Complexity

What does communication complexity have to do with extended formulations? To forge a connection, we need to show that an extended formulation with few inequalities is somehow useful for solving hard communication problems. While this course includes a number of clever connections between communication complexity and various computational models, this connection to extended formulations is perhaps the most surprising and ingenious one of them all. Superficially, extended formulations with few inequalities can be thought of as “compressed descriptions” of a polytope, and communication complexity is generally useful for ruling out compressed descriptions of various types. It is not at all obvious that this vague intuition can be turned into a formal connection, let alone one that is useful for proving non-trivial impossibility results.

#### 3.1 Faces and Facets

We discuss briefly some preliminaries about polytopes. Let $P$ be a polytope in variables $x \in \mathbb{R}^n$. By definition, an extended formulation of $P$ is a set of the form

$$Q = \{(x, y) : Cx + Dy \leq d\},$$

where $x$ and $y$ are the original and auxiliary variables, respectively, such that

$$\{x : \exists y \text{ s.t. } (x, y) \in Q\} = P.$$

This is, projecting $Q$ onto the original variables $x$ yields the original polytope $P$. The extended formulation of the permutahedron described in Section 1.2 is a canonical example; The size of the extended formulation is the number of inequalities.\(^7\)

Recall that $x \in P$ is a vertex if it cannot be written as a non-trivial convex combination of other points $P$. A supporting hyperplane of $P$ is a vector $a \in \mathbb{R}^n$ and scalar $b \in \mathbb{R}$

\(^7\)There is no need to keep track of the number of auxiliary variables — there is no point in having an extended formulation of this type with more variables than inequalities (see Exercises).
such that $a^T x \leq b$ for all $x \in P$. Every supporting hyperplane $a, b$ induces a face of $P$, defined as $\{x \in P : a^T x = b\}$ — the intersection of the boundaries of $P$ and of the the halfspace defined by the supporting hyperplane. Note that a face is generally induced by many different supporting hyperplanes. The empty set is consider a face. Note also that faces are nested — in three dimensions, there are vertices, edges, and sides. In general, if $f$ is a face of $P$, then the vertices of $f$ are precisely the vertices of $P$ that are contained in $f$.

A facet of $P$ is a maximal face — a face that is not strictly contained in any other face. Provided $P$ has a non-empty interior, its facets are $(n-1)$-dimensional.

There are two different types of finite descriptions of a polytope, and it is useful to go back and forth between them. First, a polytope $P$ equals the convex hull of its vertices. Second, $P$ is the intersection of the halfspaces that define its facets.

### 3.2 Yannakakis’s Lemma

What good is a small extended formulation? We next make up a contrived communication problem for which small extended formulations are useful. For a polytope $P$, in the corresponding $\text{FACE-VERTEX}(P)$ problem, Alice gets a face $f$ of $P$ (in the form of a supporting hyperplane $a, b$) and Bob gets a vertex $v$ of $P$. The function $FV(f, v)$ is defined as 1 if $v$ does not belong to $f$, and 0 if $v \in f$. Equivalently, $FV(f, v) = 1$ if and only if $a^T v < b$, where $a, b$ is a supporting hyperplane that induces $f$. Polytopes in $n$ dimensions generally have an exponential number of faces and vertices. Thus, trivial protocols for $\text{FACE-VERTEX}(P)$, where one party reports their input to the other, can have communication cost $\Omega(n)$.

A key result is the following.

**Lemma 3.1 (Yannakakis’s Lemma [?])** If the polytope $P$ admits an extended formulation $Q$ with $r$ inequalities, then the nondeterministic communication complexity of $\text{FACE-VERTEX}(P)$ is at most $\log_2 r$.

That is, if we can prove a linear lower bound on the nondeterministic communication complexity of the $\text{FACE-VERTEX}(P)$ game, then we have ruled out any subexponential-size extended formulation of $P$.

Sections 3.3 and 3.4 give two different proof sketches of Lemma 3.1. These are roughly equivalent, with the first emphasizing the geometric aspects (following [?]) and the second the algebraic aspects (following [?]). In Section 4 we put Lemma 3.1 to use and prove strong lower bounds for a concrete polytope.

Remarkably, Yannakakis [?] did not give any applications of his lemma — the lower bounds for extended formulations in [?] are for “symmetric” formulations and proved via direct arguments. Lemma 3.1 was suggested in [?] as a potentially useful tool for more general impossibility results, and finally in the past five years (beginning with [?]) this prophecy has come to pass.

---

8Proofs of all of these statements are elementary but outside the scope of this lecture; see e.g. [?] for details.
3.3 Proof Sketch of Lemma 3.1: A Geometric Argument

Suppose $P$ admits an extended formulation $Q = \{(x, y) : Cx + Dy \leq d\}$ with only $r$ inequalities. Both $P$ and $Q$ are known to Alice and Bob before the protocol begins. A first idea is for Alice, who is given a face $f$ of the original polytope $P$, to tell Bob the name of the “corresponding face” of $Q$. Bob can then check whether or not his “corresponding vertex” belongs to the named face or not, thereby computing the function.

Unfortunately, knowing that $Q$ is defined by $r$ inequalities only implies that it has at most $r$ facets — it can have a very large number of faces. Thus Alice can no more afford to write down an arbitrary face of $Q$ than a face of $P$.

We use a third-party prover to name a suitable facet of $Q$ than enables Alice and Bob to compute the $\text{FACE-VERTEX}(P)$ function; since $Q$ has at most $r$ facets, the protocol’s communication cost is only $\log_2 r$ as desired.

Suppose the prover wants to convince Alice and Bob that Bob’s vertex $v$ of $P$ does not belong to Alice’s face $f$ of $P$. If the prover can name a facet $f^*$ of $Q$ such that:

(i) there exists $y_v$ such that $(v, y_v) \not\in f^*$;

(ii) for every $(x, y) \in Q$ with $x \in f$, $(x, y) \in f^*$.

Such a facet $f^*$ proves that $v \not\in f$. Moreover, given $f^*$, Alice and Bob can verify (i) and (ii), respectively, without any communication.

All that remains to prove is that, when $v \not\in f$, there exists a facet $f^*$ of $Q$ such that (i) and (ii) hold. First consider the inverse image of $f$ in $Q$, $\tilde{f} = \{(x, y) \in Q : x \in f\}$. Similarly, define $\tilde{v} = \{(v, y) \in Q\}$. Since $v \not\in f$, $\tilde{f}$ and $\tilde{v}$ are disjoint subsets of $Q$. It is not difficult to prove that $\tilde{f}$ and $\tilde{v}$, as inverse images of faces under a linear map, are faces of $Q$ (exercise). An intuitive but non-trivial fact is that every face of a polytope is the intersection of the facets that contain it.\(^{9}\) Thus, for every vertex $v^*$ of $Q$ that is contained in $\tilde{v}$ (and hence not in $\tilde{f}$) — and since $\tilde{v}$ is non-empty, there is at least one — we can choose a facet $f^*$ of $Q$ that contains $\tilde{f}$ (property (ii)) but excludes $v^*$ (property (i)). This concludes the proof sketch of Lemma 3.1.

3.4 Proof Sketch of Lemma 3.1: An Algebraic Argument

The next proof sketch of Lemma 3.1 is a bit longer but introduces some of the most important concepts in the study of extended formulations.

The slack matrix of a polytope $P$ has rows indexed by faces $F$ and column indexed by vertices $V$. We identify each face with a canonical supporting hyperplane $a, b$. Entry $S_{fv}$ of the slack matrix is defined as $b - a^T v$, where $a, b$ is the supporting hyperplane corresponding to the face $f$. Observe that all entries of $S$ are nonnegative. Define the support supp($S$) of the slack matrix $S$ as the matrix with 1-entries wherever $S$ has positive entries, and 0-entries wherever $S$ has 0-entries. Observe that supp($S$) is a property only of the polytope.

\(^{9}\)This follows from Farkas’s Lemma, or equivalently the Separating Hyperplane Theorem. See [?] for details.
P, independent of the choices of the supporting hyperplanes for the faces of P. Observe also that \(\text{supp}(S)\) is precisely the answer matrix for FACE-VERTEX(P) problem for the polytope P.

We next identify a sufficient condition for FACE-VERTEX(P) to have low nondeterministic communication complexity; later we explain why the existence of a small extended formulation implies this sufficient condition. Suppose the slack matrix \(S\) has nonnegative rank \(r\), meaning it is possible to write \(S = TU\) where \(T\) is a \(|F| \times r\) nonnegative matrix and \(U\) is a \(r \times |V|\) nonnegative matrix (Figure ??).\(^{10}\) Equivalently, suppose we can write \(S\) as the sum of \(r\) outer products of nonnegative vectors (indexed by \(F\) and \(V\)):

\[
S = \sum_{j=1}^{r} \alpha_j \cdot \beta_j^T, \tag{8}
\]

where the \(\alpha_j\)'s correspond to the columns of \(T\) and the \(\beta_j\)'s to the rows of \(U\).

We claim that if the slack matrix \(S\) of a polytope \(P\) has nonnegative rank \(r\), then there is nondeterministic communication protocol for FACE-VERTEX(P) with cost at most \(\log_2 r\). As usual, Alice and Bob can agree to decomposition (8) in advance. A key observation is that, by inspection of (8), \(S_{fv} > 0\) if and only if there exists some \(j \in \{1, 2, \ldots, r\}\) with \(\alpha_{fj}, \beta_{jv} > 0\). (We are using here that everything is nonnegative and so no cancellations are possible.) Equivalently, the supports of the outer products \(\alpha_j \cdot \beta_j^T\) can be viewed as a covering of the 1-entries of \(\text{supp}(S)\) by \(r\) 1-rectangles. Given this observation, the protocol for FACE-VERTEX(P) should be clear.

1. The prover announces an index \(j \in \{1, 2, \ldots, r\}\).
2. Alice accepts if and only if the \(f\)th component of \(\alpha_j\) is strictly positive.
3. Bob accepts if and only if the \(v\)th component of \(\beta_j\) is strictly positive.

The communication cost of the protocol is clearly \(\log_2 r\). The key observation above implies that there is a proof (i.e., an index \(j \in \{1, 2, \ldots, r\}\)) accepted by both Alice and Bob if and only if Bob’s vertex \(v\) belongs to Alice’s face \(f\).

It remains to prove that, whenever a polytope \(P\) admits an extended formulation with a small number of inequalities, its slack matrix admits a low-rank nonnegative matrix factorization.\(^{11}\) We’ll show this by exhibiting nonnegative \(r\)-vectors \(\lambda_f\) (for all faces \(f\) of \(P\)) and \(\mu_v\) (for all vertices \(v\) of \(P\)) such that \(S_{fv} = \lambda_f^T \mu_v\) for all \(f\) and \(v\). In terms of Figure ??, the \(\lambda_f\)'s and \(\mu_v\)'s correspond to the rows of \(T\) and columns of \(U\), respectively.

The next is to understand better how an extended formulation \(Q = \{(x, y) : Cx + Dy \leq d\}\) must be related to original polytope \(P\). Given that projecting \(Q\) onto the variables \(x\) yields

\(^{10}\)This is called a nonnegative matrix factorization. It is the analog of the singular value decomposition (SVD), but with the extra constraint that the factors are nonnegative matrices. It obviously only makes sense to ask for such decompositions for nonnegative matrices (like \(S\)).

\(^{11}\)The converse also holds, and might well be the easier direction to anticipate. See the Exercises for details.
it must be that every supporting hyperplane of $P$ is logically implied by the inequalities that define $Q$. To see one way this can happen, suppose there is a non-negative $r$-vector $\lambda \in \mathbb{R}^r_+$ with the following properties:

(P1) $\lambda^T C = a^T$;  
(P2) $\lambda^T D = 0$;  
(P3) $\lambda^T d = b$.

(P1)–(P3) imply that, for every $(x, y)$ in $Q$ (and so with $Cx + Dy \leq d$), we have

$$\lambda^T C x + \lambda^T Dy \leq \lambda^T d$$

and hence $a^T x \leq b$ (no matter what $y$ is).

Nonnegative linear combinations $\lambda$ of the constraints of $Q$ that satisfy (P1)–(P3) are one way in which the constraints of $Q$ imply constraints on the values of $x$ in the project of $Q$. A straightforward application of Farkas’s Lemma (see e.g. [?]) implies that such nonnegative linear combinations are the only way in which the constraints of $Q$ imply constraints on the projection of $Q$.\(^{12}\) Put differently, whenever $a^T x \leq b$ is a supporting hyperplane of $P$, there exists a nonnegative linear combination $\lambda$ that proves it (i.e., that satisfies (P1)–(P3)). This clarifies what the extended formulation $Q$ really accomplishes: ranging over all $\lambda \in \mathbb{R}^r_+$ satisfying (P2) generates all of the supporting hyperplanes $a, b$ of $P$ (with $a$ and $b$ arising as $\lambda^T C$ and $\lambda^T d$, respectively).

To define the promised $\lambda_f$’s and $\mu_v$’s, fix a face $f$ of $P$ with supporting hyperplane $a^T x \leq b$. Since $Q$ projection does not include any points not in $P$, the constraints of $Q$ imply this supporting hyperplane. By the previous paragraph, we can choose a nonnegative vector $\lambda_f$ so that (P1)–(P3) hold.

Now fix a vertex $v$ of $P$. Since $Q$’s project includes every point of $P$, there exists a choice of $y_v$ such that $(v, y_v) \in Q$. Define $\mu_v \in \mathbb{R}^r_+$ as the slack in $Q$’s constraints at the point $(v, y_v)$:

$$\mu_v = d - Cv - Dy_v.$$

Since $(v, y_v) \in Q$, $\mu_v$ is a nonnegative vector.

Finally, for every face $f$ of $P$ and vertex $v$ of $P$, we have

$$\lambda_f^T \mu_v = \lambda_f^T d - \lambda_f^T Cv - \lambda_f^T Dy_v = b - a^T v = S_{fv},$$

as desired. This completes the second proof of Lemma 3.1.

---

\(^{12}\)Farkas’s Lemma is sometimes phrased as the Separating Hyperplane Theorem. It can also be thought of as the feasibility version of strong linear programming duality.
4 A Lower Bound for the Correlation Polytope

4.1 Overview

Lemma 3.1 reduces the task of proving a lower bound on the size of extended formulations of a polytope $P$ to proving lower bounds on the nondeterministic communication complexity of $\text{FACE-VERTEX}(P)$. The case study of the permutahedron (Section 1.2) serves as a cautionary tale here: the communication complexity of $\text{FACE-VERTEX}(P)$ is surprisingly low for some complex-seeming polytopes, so proving strong lower bounds, when they exist, typically requires work and a detailed understanding of the particular polytope of interest.

Fiorini et al [?] were the first to use Yannakakis’s Lemma to prove lower bounds on the size of extended formulations of interesting polytopes.\textsuperscript{13} We follow the proof plan of [?], which has two steps.

1. First, we exhibit a polytope that is tailor-made for proving a nondeterministic communication complexity lower bound on the corresponding $\text{FACE-VERTEX}(P)$ problem, via a reduction from $\text{DISJOINTNESS}$. We’ll prove this step in full.

2. Second, we extend the consequent lower bound on the size of extended formulations to other problems, such as the Traveling Salesman Problem (TSP), via reductions. These reductions are bread-and-butter $NP$-completeness-style reductions; see the Exercises for more details.

This two-step plan does not seem sufficient to resolve the motivating problem mentioned in Section ??, the non-bipartite matching problem. Indeed, we fully expect all extended formulations of the convex hull of the characteristic vectors of solutions to an $NP$-hard problem like TSP to be exponential; otherwise we could use linear programming to obtain a subexponential-time algorithm, an unlikely result. Since the non-bipartite matching problem is polynomial-time solvable, it’s less clear what to expect. Rothvoss [?] proved that every extended formulation of the convex hull of the perfect matchings of the complete graph has exponential size.\textsuperscript{14} The techniques in [?] are more sophisticated variations of the tools covered in this lecture — a reader of these notes is well-positioned to move on to the proof in [?].

4.2 Preliminaries

We describe a polytope for which it’s relatively easy to prove nondeterministic communication complexity lower bounds for the corresponding $\text{FACE-VERTEX}(P)$ problem. The polytope was studied earlier for other reasons [?].

\textsuperscript{13}This paper won the Best Paper Award at STOC ’12.

\textsuperscript{14}This paper won the Best Paper Award at STOC ’14.
Given a 0-1 $n$-bit vector $x$, we consider corresponding (symmetric and rank-1) outer product $xx^T$. For example, if $x = 10101$, then

\[
xx^T = \begin{pmatrix}
1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1
\end{pmatrix}
\]

For a positive integer $n$, we define COR as the convex hull of all $2^n$ such vectors $xx^T$ (ranging over $x \in \{0, 1\}^n$). This is a polytope in $\mathbb{R}^{n^2}$, and its vertices are precisely the points $xx^T$ with $x \in \{0, 1\}^n$.

Our goal is to prove the following result.

**Theorem 4.1** ([?]) The nondeterministic communication complexity of Face-Vertex(COR) is $\Omega(n)$.

This lower bound is clearly the best possible (up to a constant factor), since Bob can communicate his vertex to Alice using only $n$ bits (by specifying the appropriate $x \in \{0, 1\}^n$).

Theorem 4.1 then implies that every extended formulation of the COR polytope requires $2^{\Omega(n)}$ inequalities, no matter how many auxiliary variables are added. Note the dimension $d$ is $\Theta(n^2)$, so this lower bound has the form $2^{\Theta(\sqrt{n})}$.

Elementary reductions (see the Exercises) translate this extension complexity lower bound for the COR polytope to a lower bound of $2^{\Omega(\sqrt{n})}$ for extended formulations of the convex hull of characteristic vectors of $n$-point traveling salesman tours.

### 4.3 Some Faces of the Correlation Polytope

Next we establish a key connection between certain faces of the correlation polytope and inputs to DISJOINTNESS. Through, $n$ is a fixed positive integer.

**Lemma 4.2** ([?]) For every subset $S \subseteq \{1, 2, \ldots, n\}$, there is a face $f_S$ of COR such that: for every $R \subseteq \{1, 2, \ldots, n\}$ with characteristic vector $x_R$ and corresponding vertex $v_R = x_Rx_R^T$ of COR,

\[v_R \in f_S \quad \text{if and only if} \quad |S \cap R| = 1.\]

That is, among the faces of COR are $2^n$ faces that encode the “unique intersection property” for each of the $2^n$ subsets $S$ of $\{1, 2, \ldots, n\}$. Note that for a given $S$, the sets $R$ with $|S \cap R|$ can be generated by (i) first picking a element of $S$; (ii) picking a subset of $\{1, 2, \ldots, n\} \setminus S$. Thus if $|S| = k$, there are $k2^{n-2}$ sets $R$ with which it has a unique intersection.

Lemma 4.2 is kind of amazing, but also not too hard to prove.

**Proof of Lemma 4.2:** For every $S \subseteq \{1, 2, \ldots, n\}$, we need to exhibit a supporting hyperplane $a^Tx \leq b$ such that $a^Tv_R = b$ if and only of $|S \cap R| = 1$, where $v_R$ denotes $x_Rx_R^T$ and $x_R$ the characteristic vector of $R \subseteq \{1, 2, \ldots, n\}$. 

---

13
Fix $S \subseteq \{1, 2, \ldots, n\}$. We develop the appropriate supporting hyperplane, in variables $y \in \mathbb{R}^{n^2}$ over several small steps.

1. For clarity, let’s start in the wrong space, with variables $z \in \mathbb{R}^n$ rather than $y \in \mathbb{R}^{n^2}$. Here $z$ is meant to encode the characteristic vector of a set $R \subseteq \{1, 2, \ldots, n\}$. One sensible inequality to start with is

$$\sum_{i \in S} z_i - 1 \geq 0.$$  \hfill (9)

For example, if $S = \{1, 3\}$, then this constraint reads $z_1 + z_3 - 1 \geq 0$.

The good news is that for 0-1 vectors $x_R$, this inequality is satisfied with equality if and only if $|S \cap R| = 1$. The bad news is that it does not correspond to a supporting hyperplane: if $|S \cap R| \geq 2$, then $x_R$ violates the inequality. How can we change the constraint so that it holds with equality for $x_R$ with $|S \cap R| = 1$ and also holds for all $R$?

2. One crazy idea is to square the left-hand side of (9):

$$\left(\sum_{i \in S} z_i - 1\right) \geq 0.$$  \hfill (10)

For example, if $S = \{1, 3\}$, then the constraint reads (after expanding) $z_1^2 + z_3^2 + 2z_1z_3 - 2z_1 - 2z_3 + 1 \geq 0$.

The good news is that every 0-1 vector $x_R$ satisfies this inequality, and equality holds if and only if $|S \cap R| = 1$. The bad news is that the constraint is non-linear and hence does not correspond to a supporting hyperplane.

3. The obvious next idea is to “linearize” the previous constraint. Wherever the constraint has a $z_i^2$ or a $z_i$, we replace it by a variable $y_{ii}$ (note these partially cancel out). Wherever the constraint has a $2z_i z_j$ (and notice for $i \neq j$ these always come in pairs), we replace it by a $y_{ij} + y_{ji}$. Formally, the constraint now reads

$$-\sum_{i \in S} y_{ii} + \sum_{i \neq j \in S} y_{ij} + 1 \geq 0.$$  \hfill (11)

Note that the new variable set is $y \in \mathbb{R}^{n^2}$. For example, if $S = \{1, 3\}$, then the new constraint reads $y_{13} + y_{31} - y_{11} - y_{33} \geq -1$.

A first observation is that, for $y$’s that are symmetric and rank-1, with $y = zz^T$ (hence $y_{ij} = z_i \cdot z_j$ for $i, j \in \{1, 2, \ldots, n\}$), the left-hand sides of (10) and (11) are the same by definition. Thus, if in addition $y$ can be written as $x_R x_R^T$ for $x \in \{0, 1\}^n$, then $y$ satisfies the (linear) inequality (11), and equality holds if and only if $|S \cap R| = 1$. 

14
We have shown that, for every $S \subseteq \{1, 2, \ldots, n\}$, the linear inequality (11) is satisfied by every vector $y \in \mathbb{R}^n$ of the form $y = x_Rx^T_R$ with $x \in \{0, 1\}^n$. Since COR is by definition the convex hull of such vectors, every point of COR satisfies (11). This inequality is therefore a supporting hyperplane, and the face it induces contains precisely those vertices of the form $x_Rx^T_R$ with $|S \cap R| = 1$. This completes the proof. ■

4.4 **Face-Vertex(COR) and Unique-Disjointness**

In the **Face-Vertex(COR) problem**, Alice receives a face $f$ of COR and Bob a vertex $v$ of COR. In the 1-inputs, $v \notin f$; in the 0-inputs, $v \in F$. Let’s make the problem only easier by restricting Alice’s possible inputs to the $2^n$ faces (one per subset $S \subseteq \{1, 2, \ldots, n\}$) identified in Lemma 4.2. In the corresponding matrix $M_U$ of this function, we can index the rows by subsets $S$. Since every vertex of COR has the form $y = x_Rx^T_R$ for $R \subseteq \{1, 2, \ldots, n\}$, we can index the columns of $M_U$ by subsets $R$. By Lemma 4.2, the entry $(S, R)$ of the matrix $M_U$ is 1 if $|S \cap R| \neq 1$ and 0 if $|S \cap R| = 1$. That is, the 0-entries of $M_U$ correspond to pairs $(S, R)$ that intersect in a unique element.

There is clearly a strong connection between the matrix $M_U$ above and the analogous matrix $M_D$ for **Disjointness**. They differ on entries $(S, R)$ with $|S \cap R| \geq 2$: these are 0-entries of $M_D$ but 1-entries of $M_U$. In other words, $M_U$ is the matrix corresponding to the communication problem $\neg$**Unique-Intersection**: do the inputs $S$ and $R$ fail to have a unique intersection?

The closely related **Unique-Disjointness** problem is a “promise” version of **Disjointness**. The task here is to distinguish between:

1. inputs $(S, R)$ of Fwith $|S \cap R| = 0$;
2. inputs $(S, R)$ of Fwith $|S \cap R| = 1$.

For inputs that fall into neither case (with $|S \cap R| > 1$), the protocol is off the hook — any output is considered correct. Since a protocol that solves **Unique-Disjointness** has to do only less than one that solves $\neg$**Unique-Intersection**, communication complexity lower bounds for former problem apply immediate to the latter.

We summarize the discussion of this section in the following proposition.

**Proposition 4.3** ([?]) The nondeterministic communication complexity of **Face-Vertex(COR)** is at least that of **Unique-Disjointness**.

4.5 **A Lower Bound for Unique-Disjointness**

4.5.1 The Goal

One final step remains in our proof of Theorem 4.1, and hence of our lower bound on the size of extended formulations of the correlation polytope.

**Theorem 4.4** ([?], [?]) The nondeterministic communication complexity of **Unique-Disjointness** is $\Omega(n)$. 

15
4.5.2 DISJOINTNESS Revisited

As a warm-up, we revisit the standard DISJOINTNESS problem. Recall that, in Lecture #4, we proved the nondeterministic communication complexity of DISJOINTNESS is $n$ using a fooling set argument. Next we prove a slightly weaker lower bound, via an argument that generalizes to UNIQUE-DISJOINTNESS.

The first claim is that, of the $2^n \times 2^n = 4^n$ possible inputs in DISJOINTNESS, exactly $3^n$ of them are 1-inputs. The reason is that the following procedure, which makes $n$ 3-way choices, generates every 1-input exactly once: independently for each coordinate $i = 1, 2, \ldots, n$, choose between the options (i) $x_i = y_i = 0$; (ii) $x_i = 1$ and $y_i = 0$; and (iii) $x_i = 0$ and $y_i = 1$.

The second claim is that every 1-rectangle — every subset $A$ of rows of $M_D$ and $B$ of columns of $M_D$ such that $A \times B$ contains only 1-inputs — has size at most $2^n$. To prove this, let $R = A \times B$ be a 1-rectangle. We assert that, for every coordinate $i = 1, 2, \ldots, n$, either (i) $x_i = 0$ for all $x \in A$ or (ii) $y_i = 0$ for all $y \in B$. That is, every coordinate has, for at least one of the two parties, a “forced zero” in $R$. For if neither (i) nor (ii) hold for a coordinate $i$, then since $R$ is a rectangle (and hence closed under “mix and match”) we can choose $(x, y) \in R$ with $x_i = y_i = 1$; but this is a 0-input and $R$ is a 1-rectangle. This assertion implies that the following procedure, which makes $n$ 2-way choices, generates every 1-input of $R$ (and possibly other inputs as well): independently for each coordinate $i = 1, 2, \ldots, n$, set $x_i = 0$ (in case (i)) or $y_i = 0$ (in case (ii)).

These two claims imply that every covering of the 1-inputs by 1-rectangles requires at least $(3/2)^n$ rectangles. Proposition 2.2 then implies a lower bound of $\Omega(n)$ on the nondeterministic communication complexity of DISJOINTNESS.

4.5.3 Proof of Theorem 4.4

Recall that the 1-inputs $(x, y)$ of UNIQUE-DISJOINTNESS are the same as those of $F$ (for each $i$, either $x_i = 0$, $y_i = 0$, or both). Thus, there are still exactly $3^n$ 1-inputs. The 0-inputs $(x, y)$ of UNIQUE-DISJOINTNESS are those with $x_i = y_i = 1$ in exactly one coordinate $i$. We call all other inputs, where the promise fails to hold, *-inputs. By a 1-rectangle, we now mean that a rectangle that does not contain any 0-inputs (*-inputs are fine).

**Lemma 4.5** Every 1-rectangle contains at most $2^n$ 1-inputs.

As with the argument for $F$, Lemma 4.5 completes the proof of Theorem 4.4: since there are $3^n$ 1-inputs and at most $2^n$ per 1-rectangle, every covering by 1-rectangles requires at least $(3/2)^n$ rectangles, implying that the nondeterministic communication complexity of UNIQUE-DISJOINTNESS is $\Omega(\log n)$.

Why is Lemma 4.5 harder than before? We can no longer easily argue that, in a rectangle $R = A \times B$, for each coordinate $i$, either $x_i = 0$ for all $x \in A$ or $y_i = 0$ for all $y \in B$. Assume the opposite no longer yields a contraction: exhibiting $x \in A$ and $y \in B$ with $x_i = y_i = 1$ does not necessarily contradict the fact that $R$ is a 1-box, since $(x, y)$ might be a *-input.
Proof of Lemma 4.5: The proof is one of those slick inductions that you can’t help but sit back and admire.

We claim, by induction on \( k = 0, 1, 2, \ldots, n \), that if \( R = A \times B \) is a 1-rectangle for which all \( xinA \) and \( yinB \) have 0s in their last \( n - k \) coordinates, then the number of 1-inputs in \( R \) is at most \( 2^k \). The lemma is equivalent to the case of \( k = n \). The base case \( k = 0 \) holds, because in this case the only possible input in \( R \) is \((0, 0)\).

For the inductive step, fix a 1-rectangle \( R = A \times B \) in which the last \( n - k \) coordinates of all \( x \in A \) and all \( yinB \) are 0. To simplify notation, from here on we ignore the last \( n - k \) coordinates of all inputs (they play no role in the argument).

Intuitively, we need to somehow “zero out” the \( k \)th coordinate of all inputs in \( R \) so that we can apply the inductive hypothesis. This motivates focusing on the \( k \)th coordinate, and we’ll often write inputs \( x \in A \) and \( y \in B \) as \( x'a \) and \( y'b \), respectively, with \( x', y' \in \{0, 1\}^{k-1} \) and \( a, b \in \{0, 1\} \). (Recall we’re ignoring that last \( n - k \) coordinates, which are now always zero.)

First observe that, whenever \((x'a, y'b)\) is a 1-input, we cannot have \( a = b = 1 \). Also:

\((^*)\) If \((x'a, y'b) \in R\) is a 1-input, then \( R \) cannot contain both the inputs \((x'0, y'1)\) and \((x'1, y'0)\).

For otherwise, \( R \) would also contain the 0-input \((x'1, y'1)\), contradicting that \( R \) is a 1-rectangle. (Since \((x'a, y'b)\) is a 1-input, the unique coordinate of \((x'1, y'1)\) with a 1 in both inputs is the \( k \)th coordinate.)

The plan for the rest of the proof is to define two sets \( S_1, S_2 \) of 1-inputs — not necessarily rectangles — such that:

(P1) the number of 1-inputs in \( S_1 \) and \( S_2 \) combined is at least that in \( R \);
(P2) the inductive hypothesis applies to \( \text{rect}(S_1) \) and \( \text{rect}(S_2) \), where \( \text{rect}(S) \) denotes the smallest rectangle containing a set \( S \) of inputs.\(^{15}\)

We can find sets \( S_1, S_2 \) with properties (P1),(P2), then we are done: by the inductive hypothesis, the \( \text{rect}(S_i) \)'s have at most \( 2^{k-1} \) 1-inputs each, the \( S_i \)'s are only smaller, and hence (by (P1)) \( R \) has at most \( 2^k \) inputs, as required.

We define the sets in two steps, focusing first on property (P1). Recall that every 1-input \((x, y) \in R\) has the form \((x'1, y'0)\), \((x'0, y'1)\), or \((x'0, y'0)\). We put all 1-inputs of the first type into a set \( S'_1 \), and all 2-inputs of the second type into a set \( S'_2 \). When placing inputs of the third type, we want to avoid putting two inputs of the form \((x'a, y'b)\) with the same \( x' \) and \( y' \) into the same set (this would create problems in the inductive step). So, for an input \((x'0, y'0) \in R\), we put it in \( S'_1 \) if and only if the input \((x'1, y'0)\) was not already put in \( S'_1 \); and we put it in \( S'_2 \) if and only if the input \((x'0, y'1)\) was not already put in \( S'_2 \). Crucially, observation \((^*)\) implies that \( R \) cannot contain two 1-inputs of the form \((x'1, y'0)\) and \((x'0, y'1)\), so the 1-input \((x'0, y'0)\) is placed in at least one of the sets \( S'_1, S'_2 \). (It is

\(^{15}\)Equivalently, the closure of \( S \) under the “mix and match” operation on pairs of inputs. Formally, \( \text{rect}(S) = X(S) \times Y(S) \), where \( X(S) = \{x : (x, y) \in S \text{ for some } y\} \) and \( Y(S) = \{y : (x, y) \in S \text{ for some } x\} \).
placed in both if \( R \) contains neither \((x'1, y'0)\) nor \((x'0, y'1)\).) By construction, the sets \( S'_1 \) and \( S'_2 \) satisfy property (P1).

We next make several observations about \( S'_1 \) and \( S'_2 \). By construction, (***) for each \( i = 1, 2 \) and \( x', y' \in \{0, 1\}^{k-1} \), there is at most one input of \( S'_i \) of the form \((x'a, y'b)\).

Also, since \( S'_1, S'_2 \) are subsets of the rectangle \( R \), \( \text{rect}(S'_1), \text{rect}(S'_2) \) are also subsets of \( R \). Since \( R \) is a 1-rectangle, so are \( \text{rect}(S'_1), \text{rect}(S'_2) \). Also, since every input \((x, y)\) of \( S'_i \) (and hence \( \text{rect}(S'_i) \)) has \( y_k = 0 \) (for \( i = 1 \)) or \( x_k = 0 \) (for \( i = 2 \)), the \( k \)th coordinate contributes nothing to the intersection of any inputs of \( \text{rect}(S'_1) \) or \( \text{rect}(S'_2) \).

Now obtain \( S_i \) from \( S'_i \) (for \( i = 1, 2 \)) by zeroing out the \( k \)th coordinate of all inputs. Since the \( S'_i \)'s only contain 1-inputs, the \( S_i \)'s only contain 1-inputs. Since property (***) implies that \(|S_i| = |S'_i|\) for \( i = 1, 2 \), we conclude that property (P1) holds also for \( S_1, S_2 \).

Moving on to property (P2), since \( \text{rect}(S'_1), \text{rect}(S'_2) \) contain no 0-inputs and contain only inputs with no intersection in the \( k \)th coordinate, \( \text{rect}(S_1), \text{rect}(S_2) \) contain no 0-inputs.\(^{16}\) Finally, since all inputs of \( S_1, S_2 \) have zeroes in their final \( n-k+1 \) coordinates, so do all inputs of \( \text{rect}(S_1), \text{rect}(S_2) \). The inductive hypothesis applies to \( \text{rect}(S_1) \) and \( \text{rect}(S_2) \), so each of them has at most \( 2^{k-1} \) 1-inputs. This implies the inductive step and completes the proof. ■

\(^{16}\)The concern is that zeroing out an input in the \( k \)th coordinate turns some *-input (with intersection size 2) into a 0-input (with intersection size 1); but since there were no intersections in the \( k \)th coordinate, anyways, this can’t happen.