Chapter 9
Monotone Circuits

We now consider monotone circuits, that is, circuits with fanin-2 AND and OR gates. As monotone formulas, such circuits can only compute monotone boolean functions. Recall that a boolean function f is *monotone* if $f(x) \leq f(y)$ as long as $x_i \leq y_i$ for all i. The difference from formulas is that now the fan-outs of gates may be arbitrary, not just 1. That is, a result computed at some gate can be used many times with no need to recompute it again and again. This additional feature makes the lower bounds problem more difficult.

Until 1985, the largest known lower bound on the size of such circuits for an explicit boolean function of n variables was only $4n$ (Tiekenheinrich 1984). A breakthrough was achieved in 1985 when two mathematicians from Lomonosov University in Moscow—Andreev (1985) and Razborov (1985a)—almost simultaneously proved super-polynomial lower bounds for monotone circuits.

In this chapter we present Razborov's method of approximations as well as another, simpler argument yielding exponential lower bounds even for circuits with monotone *real-valued* functions as gates.

As in the entire book, here our focus is on proving lower bounds. A comprehensive exposition of known upper bounds for monotone circuits and monotone switching networks can be found in a survey by Korshunov (2003).

9.1 Large Cliques are Hard to Detect

We will first demonstrate Razborov's method of approximations for the case of monotone circuits computing the clique function. Later, in Sect. 9.10, we describe his method in its full generality, and apply it to the perfect matching function.

The *clique function* $f_n = \text{CLIQUE}(n,k)$ has $\binom{n}{2}$ variables x_{ij}, one for each potential edge in a graph on n vertices $[n] = \{1,\ldots,n\}$; the function outputs 1 iff the associated graph contains a clique (complete subgraph) on some k vertices. The clique function is monotone because setting more edges to 1 can only increase the size of the largest clique.

S. Jukna, *Boolean Function Complexity*, Algorithms and Combinatorics 27,
DOI 10.1007/978-3-642-24508-4_9,

Theorem 9.1. (Razborov 1985a; Alon and Boppana 1987) *For* $3 \le k \le n^{1/4}$*, the monotone circuit complexity of* CLIQUE(n,k) *is* $n^{\Omega(\sqrt{k})}$.

We will analyze the behavior of circuits for f_n on two types of input graphs:

- *Positive graphs* are k-cliques, that is, graphs consisting of a clique on some k vertices and $n-k$ isolated vertices; we have $\binom{n}{k}$ such graphs and they all must be accepted by f_n.
- *Negative graphs* are $(k-1)$-cocliques formed by assigning each vertex a color from the set $\{1,2,\ldots,k-1\}$, and putting edges between those pairs of vertices with different colors; we have $(k-1)^n$ such graphs and they must be rejected by f_n. (Different colorings can lead to the same graph, but we will consider them as different for counting purposes.)

The main goal of Razborov's method is to show that, if a circuit is "too small", then it must make a lot of errors, that is, must either reject most of positive graphs or accept most of negative graphs. Circuits can be amorphous, so analyzing their behavior directly is difficult. Instead, every monotone circuit will be *approximated* by another monotone circuit of a very special type—namely, a short DNF that is tailor-made to represent collections of cliques.

Now we define these DNFs, our so-called "approximators". For a subset X of vertices, the *clique indicator* of X is the monotone boolean function $\lceil X \rceil$ of $\binom{n}{2}$ variables such that $\lceil X \rceil(E) = 1$ if and only if the graph E contains a clique on the vertices X. Note that $\lceil X \rceil$ is just a monomial

$$\lceil X \rceil = \bigwedge_{i,j \in X; i<j} x_{ij}$$

depending on only $\binom{|X|}{2}$ variables.

An (m,l)-*approximator* is an OR of at most m clique indicators, whose underlying vertex-sets each have cardinality at most l:

$$A = \bigvee_{t=1}^{r} \lceil X_t \rceil = \bigvee_{t=1}^{r} \bigwedge_{i \ne j \in X_t} x_{ij} \qquad (r \le m,\ |X_t| \le l)\,.$$

Here $l \ge 2$ and $m \ge 2$ are parameters depending only on values of k and n; the values of these parameters will be fixed later.

The main combinatorial tool used in the proof of Theorem 9.1 is the well-known *Sunflower Lemma* discovered by Erdős and Rado (1960). A *sunflower* with p petals and a core T is a collection of sets $S_1,\ldots,S_p$ such that $S_i \cap S_j = T$ for all $i \ne j$. In other words, each element belongs either to none, or to exactly one, or to *all* of the S_i (Fig. 9.1). Note that a family of pairwise disjoint sets is a sunflower (with an empty core).

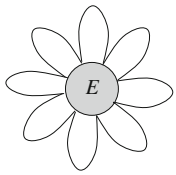

Fig. 9.1 A sunflower with eight petals

Sunflower Lemma. Let $\mathcal{F}$ be family of non-empty sets each of size at most l. If $|\mathcal{F}| > l!(p-1)^l$ then $\mathcal{F}$ contains a sunflower with p petals.

In particular, every graph with at least $2(p-1)^2+1$ edges must have p vertex-disjoint edges of a star with p edges.

Proof. We proceed by induction on l. For $l = 1$, we have more than $p-1$ points (disjoint 1-element sets), so any p of them form a sunflower with p petals (and an empty core). Now let $l \geq 2$, and take a maximal family $\mathcal{S} = \{S_1, \ldots, S_t\}$ of pairwise disjoint members of $\mathcal{F}$.

If $t \geq p$, these sets form a sunflower with $t \geq p$ petals (and empty core), and we are done.

Assume that $t \leq p-1$, and let $S = S_1 \cup \cdots \cup S_t$. Then $|S| \leq l(p-1)$. By the maximality of $\mathcal{S}$, the set S intersects every member of $\mathcal{F}$. By the pigeonhole principle, some point $x \in S$ must be contained in at least

$$\frac{|\mathcal{F}|}{|S|} > \frac{l!(p-1)^l}{l(p-1)} = (l-1)!(p-1)^{l-1}$$

members of $\mathcal{F}$. Let us delete x from these sets and consider the family

$$\mathcal{F}_x := \{F \setminus \{x\} \,:\, F \in \mathcal{F},\, x \in F\}\,.$$

By the induction hypothesis, this family contains a sunflower with p petals. Adding x to the members of this sunflower, we get the desired sunflower in the original family $\mathcal{F}$. □

9.1.1 Construction of the Approximated Circuit

Given a monotone circuit F for the clique function f_n, we will construct the approximator for F in a "bottom-up" manner, starting from the input variables. An input variable is of the form x_{ij}, where i and j are different vertices; it is equivalent to the clique indicator $\lceil\{i,j\}\rceil = x_{ij}$.

Suppose at some internal node of the circuit, say at an OR gate, the two subcircuits feeding into this gate already have their (m,l)-approximators $A = \bigvee_{i=1}^{r} \lceil X_i \rceil$ and $B = \bigvee_{i=1}^{s} \lceil Y_i \rceil$, where r and s are at most m. We could approximate this OR

gate by just $A \vee B$, but this could potentially give us a $(2m, l)$-approximator, while we want to stay at (m, l).

At this place the Sunflower Lemma comes to our rescue. To apply the Sunflower Lemma to the present situation, consider the family

$$\mathcal{F} = \{X_1, \ldots, X_r, Y_1, \ldots, Y_s\}$$

and set

$$m := l!(p-1)^l .$$

If $r + s > m$ then some p of the sets in $\mathcal{F}$ form a sunflower. We then replace these p sets by their core; this operation is called a *plucking*. Repeatedly perform such pluckings until no more are possible. The entire procedure is called the *plucking procedure*. Since the number of vertex sets decreases with each plucking, after at most $|\mathcal{F}| = r + s \leq 2m$ pluckings we will obtain an (m, l)-approximator for our OR gate, which we denote by $A \sqcup B$.

If the gate was an AND gate (not an OR gate) then forming the AND of the two approximators $A = \bigvee_{i=1}^{r} \lceil X_i \rceil$ and $B = \bigvee_{i=1}^{s} \lceil Y_i \rceil$ yields the expression $\bigvee_{i=1}^{r} \bigvee_{j=1}^{s} (\lceil X_i \rceil \wedge \lceil Y_i \rceil)$. Two reasons why this expression itself might not be an (m, l)-approximator are that the terms $\lceil X_i \rceil \wedge \lceil Y_i \rceil$ might not be clique indicators and that there can be as many as m^2 terms.

To overcome these difficulties, apply the following three steps:

1. Replace the term $\lceil X_i \rceil \wedge \lceil Y_i \rceil$ by the clique indicator $\lceil X_i \cup Y_i \rceil$;
2. Erase those clique indicators $\lceil X_i \cup Y_i \rceil$ for which $|X_i \cup Y_j| \geq l + 1$;
3. Apply the plucking procedure (described above for OR gates) to the remaining clique indicators; there will be at most m^2 pluckings.

These three steps guarantee that an (m, l)-approximator is formed; we denote it by $A \sqcap B$. (Note an "asymmetry" in the argument: AND gates need more work to approximate than OR gates.)

9.1.2 Bounding Errors of Approximation

Now fix a monotone circuit F computing $f_n = \text{CLIQUE}(n, k)$, and let F' be the approximated circuit, that is, an (m, l)-approximator of the last gate of F. We will show that

1. Every approximator (including F') must make a lot of errors, that is, disagree with f_n on many negative and positive graphs.
2. If size(F) is small, then F' cannot make too many errors.

This will already imply that size(F) must be large.

Lemma 9.2. *Every approximator either rejects all graphs or wrongly accepts at least a fraction* $1 - l^2/(k-1)$ *of all* $(k-1)^n$ *negative graphs.*

Proof. Let $A = \bigvee_{i=1}^{r} \lceil X_i \rceil$ an (m,l)-approximator, and assume that A accepts at least one graph. Then $A \geq \lceil X_1 \rceil$. A negative graph is rejected by the clique indicator $\lceil X_1 \rceil$ iff its associated coloring assigns some two vertices of X_1 the same color. We have $\binom{|X_1|}{2}$ pairs of vertices in X_1, and for each such pair at most $(k-1)^{n-1}$ colorings assign the same color. Thus, at most $\binom{|X_1|}{2}(k-1)^{n-1} \leq \binom{l}{2}(k-1)^{n-1}$ negative graphs can be rejected by $\lceil X_1 \rceil$, and hence, by the approximator A. □

Thus, every approximator (including F') must make a lot of errors. We are now going to show that, if $\text{size}(F)$ is small, then the number of errors cannot be large, implying that $\text{size}(F)$ must be large.

Lemma 9.3. *The number of positive graphs wrongly rejected by F' is at most* $\text{size}(F) \cdot m^2 \binom{n-l-1}{k-l-1}$.

Proof. We shall consider the errors introduced by the approximator of a single gate, and then apply the union bound to get the claimed upper bound on the total number of errors.

If g is an OR gate and A, B are the approximators of subcircuits feeding into this gate, then our construction of the approximator $A \sqcup B$ for g involves taking an OR $A \vee B$ (which does not introduce any errors) and then repeatedly plucking until we get down our number of clique indicators. Each plucking replaces a clique indicator $\lceil X_i \rceil$ by some $\lceil X \rceil$ with $X \subseteq X_i$ which can accept only more graphs. Hence, on positive graphs, $A \sqcup B$ produces no errors at all.

Now suppose that g is an AND gate. The first step in the transformation from $A \wedge B$ to $A \sqcap B$ is to replace $\lceil X_i \rceil \wedge \lceil Y_j \rceil$ by $\lceil X_i \cup Y_j \rceil$. These two functions behave identically on positive graphs (cliques). The second step is to erase those clique indicators $\lceil X_i \cup Y_j \rceil$ for which $|X_i \cup Y_j| \geq l+1$. For each such clique indicator, at most $N := \binom{n-l-1}{k-l-1}$ of the positive graphs are lost. Since there are at most m^2 such clique indicators, at most $m^2 N$ positive graphs are lost in the second step. The third and final step, applying the plucking procedure, only enlarges the class of accepted graphs, as noted in the previous paragraph. Summing up the three steps, at most $m^2 N$ positive graphs can be lost by approximating one AND gate. Since we have at most $\text{size}(F)$ such gates, the lemma is proved. □

Lemma 9.4. *The number of negative graphs wrongly accepted by F' is at most* $\text{size}(F) \cdot m^2 l^{2p} (k-1)^{n-p}$.

Proof. Again, we shall analyze the errors introduced at each gate.

If g is an OR gate and A, B are the approximators of subcircuits feeding into this gate, then our construction of the approximator $A \sqcup B$ for g involves taking an OR $A \vee B$ (which does not introduce any errors) and then performing at most $2m$ pluckings until we get down our number of clique indicators.

Each plucking will be shown to accept only a few additional negative graphs. Color the vertices randomly, with all $(k-1)^n$ possible colorings equally likely, and let G be the associated negative graph. Let $Z_1, \ldots, Z_p$ be the petals of a sunflower with core Z. What is the probability that $\lceil Z \rceil$ accepts G, but none of the functions

$\lceil Z_1 \rceil, \ldots, \lceil Z_p \rceil$ accept G? This event occurs iff the vertices of Z are assigned distinct colors (called a proper coloring, or PC), but every petal Z_i has two vertices colored the same. We have

$$\begin{aligned}
&\text{Prob}[Z \text{ is PC and } Z_1, \ldots, Z_p \text{ are not PC}] \\
&\quad \leq \text{Prob}[Z_1, \ldots, Z_p \text{ are not PC} | Z \text{ is PC}] \\
&\quad = \prod_{i=1}^{p} \text{Prob}[Z_i \text{ is not PC} | Z \text{ is PC}] \\
&\quad \leq \prod_{i=1}^{p} \text{Prob}[Z_i \text{ is not PC}] \\
&\quad \leq \binom{l}{2}^p \cdot (k-1)^{-p} \leq l^{2p}(k-1)^{-p}\,.
\end{aligned}$$

The first inequality holds by the definition of the conditional probability. The second line holds because the sets $Z_i \setminus Z$ are disjoint and hence the events are independent. The third line holds because the event "Z_i is not a clique" is less likely to happen given the fact that $Z \subseteq Z_i$ is a clique. The fourth line holds because Z_i is not properly colored iff two vertices of Z_i get the same color.

Thus to the class of wrongly accepted negative graphs each plucking adds at most $l^{2p}(k-1)^{n-p}$ new graphs. There are at most $2m$ pluckings, so the total number of negative graphs wrongly accepted when approximating the gate OR g is at most $2ml^{2p}(k-1)^{n-p}$.

Next consider the case when g is an AND gate. In the transformation from $A \wedge B$ to $A \sqcap B$, the first step introduces no new violations, since $\lceil X_i \rceil \wedge \lceil Y_j \rceil \geq \lceil X_i \cup Y_j \rceil$. Only the third step, the plucking procedure, introduces new violations. This step was analyzed above; the only difference is that there can be m^2 pluckings instead of just $2m$. This settles the case of AND gates, thus completing the proof. □

Proof of Theorem 9.1. Set $l = \lfloor \sqrt{k-1}/2 \rfloor$ and $p = \Theta(\sqrt{k} \log n)$; recall that $m = l!(p-1)^l \leq (pl)^l$. Let F be a monotone circuit that computes CLIQUE(n,k). By Lemma 9.2, the approximator F' of F is either identically 0 or outputs 1 on at least a $(1 - l^2/(k-1)) \geq \frac{1}{2}$ fraction of all $(k-1)^n$ negative graphs. If the former case holds, then apply Lemma 9.3 to obtain

$$\text{size}(F) \cdot m^2 \cdot \binom{n-l-1}{k-l-1} \geq \binom{n}{k}.$$

Since $\binom{n}{k} / \binom{n-x}{k-x} \geq (n/k)^x$, simple calculation show that in this case size(F) is $n^{\Omega(\sqrt{k})}$. If the later case holds then apply Lemma 9.4 to obtain

$$\text{size}(F) \cdot m^2 \cdot 2^{-p} \cdot (k-1)^n \geq \frac{1}{2}(k-1)^n .$$

Since $2^p = n^{\Omega(\sqrt{k})}$, in this case we again have that size(F) is $n^{\Omega(\sqrt{k})}$. □

Remark 9.5. Recently, Rossman (2010) gave lower bounds for the Clique function that apply to finding small cliques in *random graphs*. Let $G(n, p)$ denote a random graph on n vertices in which each edge appears at random and independently with probability p. Let k be a fixed natural number. It is well known that $p := n^{-2/(k-1)}$ is a threshold for appearance of k-cliques. Rossman showed that, for every constant k, no monotone circuit of size smaller than $\mathcal{O}(n^{k/4})$ can correctly compute (with high probability) the Clique function on $G(n, p)$ and on $G(n, 2p)$ *simultaneously*.

9.2 Very Large Cliques are Easy to Detect

By Theorem 9.1, we known that there exists a constant $c > 0$ such that every monotone circuit computing the clique function CLIQUE(n, k) requires at least $n^{c\sqrt{k}}$ gates. Moreover, it can be shown (see Theorem 9.19 below) that already for $k = 3$ at least $\Omega(n^3/\log^4 n)$ gates are necessary. In fact, Alon and Boppana (1987) showed that Razborov's lower bound can be improved to $\Omega((n/\log^2 n)^k)$ for any constant $k \geq 3$, and for growing k we need at least $2^{\Omega(\sqrt{k})}$ gates, as long as $k \leq (n/\log n)^{2/3}/4$. Thus, small cliques are hard to detect.

By a simple padding argument, this implies that even detecting cliques of size $n-k$ requires a super-polynomial number of gates, as long as $k \leq n/2$ grows faster than $\log^3 n$.

Proposition 9.6. *For $k \leq n/2$, every monotone circuit for* CLIQUE$(n, n-k)$ *requires $2^{\Omega(k^{1/3})}$ gates.*

Proof. Fix the integer m with $m - s = k$ where $s = \lfloor (m/\log m)^{2/3}/4 \rfloor$; hence $s = \Omega(k^{2/3})$. Then CLIQUE(m, s) is a sub-function of (that is, can be obtained by setting to 1 some variables in) CLIQUE$(n, n-k)$: just consider only the n-vertex graphs containing a fixed clique on $n - m$ vertices connected to all the remaining vertices (the rest may be arbitrary). On the other hand, according to the lower bound of Alon and Boppana (mentioned above) the function CLIQUE(m, s), and hence, also the function CLIQUE$(n, n-k)$ requires monotone circuits of size exponential in $\Omega(\sqrt{s}) = \Omega(k^{1/3})$. □

But what is the complexity of CLIQUE$(n, n-k)$ when k is very small, say, constant—can this function then be computed by a monotone circuit using substantially fewer than n^k gates? Somewhat surprisingly, for every(!) constant k, the CLIQUE$(n, n-k)$ function can be computed by a monotone circuit of size $\mathcal{O}(n^2 \log n)$. Moreover, the number of gates is polynomial, as long as $k = O(\sqrt{\log n})$. Recall that CLIQUE(n, k) requires $\Omega(n^k/\log^{2k} n)$ for every constant

k, and that already for $k = \omega(\log^3 n)$, any monotone circuit for CLIQUE$(n, n-k)$ requires a super-polynomial number of gates.

Theorem 9.7. (Andreev and Jukna 2008) *For every constant k, the function* CLIQUE$(n, n-k)$ *can be computed by a monotone DeMorgan formula containing at most $\mathcal{O}(n^2 \log n)$ gates. The number of gates remains polynomial in n as long as $k = \mathcal{O}(\sqrt{\log n})$.*

In this section we will prove Theorem 9.7. To do this, we need some preparations. First, instead of constructing a small formula for the Clique function, it will be convenient to construct a small formula for the dual function. Recall that the *dual* of a boolean function $f(x_1, \ldots, x_n)$ is the boolean function $f^*(x_1, \ldots, x_n) = \neg f(\neg x_1, \ldots, \neg x_n)$. If f is monotone, then its dual f^* is also monotone. For example,

$$(x \vee y)^* = \neg(\neg x \vee \neg y) = x \wedge y\,;$$
$$(x \wedge y)^* = \neg(\neg x \wedge \neg y) = x \vee y\,.$$

In particular, the dual of CLIQUE$(n, n-k)$ accepts a given graph G on n vertices iff G has no independent set with $n-k$ vertices, which is equivalent to $\tau(G) \geq k+1$, where $\tau(G)$ ist the vertex-cover number of G. Recall that a *vertex cover* in a graph G is a set of its vertices containing at least one endpoint of each edge; $\tau(G)$ is the minimum size of such a set. Hence, the dual of CLIQUE$(n, n-k)$ is a monotone boolean function VC(n, k) of $\binom{n}{2}$ boolean variables representing the edges of an undirected graph G on n vertices, whose value is 1 iff G does not have a vertex-cover of cardinality k.

We will construct a monotone formula for VC(n, k). Replacing OR gates by AND gates (and vice versa) in this formula yields a monotone formula for CLIQUE$(n, n-k)$, thus proving Theorem 9.7.

9.2.1 Properties of τ-Critical Graphs

A graph is *τ-critical* if removing any of its edges reduces the vertex-cover number. We will need some properties of such graphs.

Theorem 9.8. (Hajnal 1965) *In a τ-critical graph without isolated vertices every independent set S has at least $|S|$ neighbors.*

Proof. Let $G = (V, E)$ be a τ-critical graph without isolated vertices. Then G is also α-critical in that removing of any its edge increases its independence number $\alpha(G)$, that is, the maximum size of an independent set in G. An independent set T is *maximal* if $|T| = \alpha(G)$.

Let us first show that every vertex belongs to at least one maximal independent set but not to all such sets. For this, take a vertex x and an edge $e = \{x, y\}$. Remove

e from G. Since G is α-critical, the resulting graph has an independent set T of size $\alpha(G)+1$. Since T was not independent in G, $x, y \in T$. Then $T \setminus \{x\}$ is an independent set in G of size $|T \setminus \{x\}| = \alpha(G)$, that is, is a maximal independent set avoiding the vertex x, and $T \setminus \{y\}$ is a maximal independent set containing x.

Hence, if X is an arbitrary independent set in G, then the intersection of X with *all* maximal independent sets in G is empty. It remains therefore to show that, if Y is an arbitrary independent set, and S is an intersection of Y with an arbitrary number of maximal independent sets, then

$$|N(Y)| - |N(S)| \geq |Y| - |S|,$$

where $N(Y)$ is the set of all neighbors of Y, that is, the set of all vertices adjacent to at least one vertex in Y. Since an intersection of independent sets is an independent set, it is enough to prove the claim for the case when T is a maximal independent set and $S = Y \cap T$. Since clearly $N(S) \subseteq N(Y) - T$, we have

$$\begin{aligned} |N(Y)| - |N(S)| &\geq |N(Y) \cap T| \\ &= |T| - |S| - |T \setminus (Y \setminus N(Y))| \\ &= \alpha(G) - |S| + |Y| - |(T \cup Y) \setminus N(Y)| \\ &\geq |Y| - |S|, \end{aligned}$$

where the last inequality holds because the set $(T \cup Y) - N(Y)$ is independent. □

In our construction of a small circuit for the Vertex Cover function, the following consequence of this theorem will be important.

Corollary 9.9. *Every τ-critical graph G has at most $2\tau(G)$ non-isolated vertices.*

Proof. Let $G = (V, E)$ be an arbitrary τ-critical graph, and let $U \subseteq V$ be the set of non-isolated vertices of G. The induced subgraph $G' = (U, E)$ has no isolated vertices and is still τ-critical with $\tau(G') = \tau(G)$. Let $S \subseteq U$ be an arbitrary vertex-cover of G' with $|S| = \tau(G)$. The complement $T = U - S$ is an independent set. By Hajnal's theorem, the set T must have at least $|T|$ neighbors. Since all these neighbors must lie in S, the desired upper bound $|U| = |S| + |T| \leq 2|S| \leq 2\tau(G)$ on the total number of non-isolated vertices of G follows. □

Finally, we will need a fact stating that τ-critical graphs cannot have too many edges. We will derive this fact from the following more general result.

Theorem 9.10. (Bollobás 1965) *Let $A_1, \ldots, A_m$ and $B_1, \ldots, B_m$ be two sequences of sets such that $A_i \cap B_j = \emptyset$ if and only if $i = j$. Then*

$$\sum_{i=1}^{m} \binom{|A_i| + |B_i|}{|A_i|}^{-1} \leq 1\,. \tag{9.1}$$

Proof. Let X be the union of all sets $A_i \cup B_i$. If A and B are disjoint subsets of X then we say that a permutation $(x_1, x_2, \ldots, x_n)$ of X *separates* the pair (A, B) if no element of B precedes an element of A, that is, if $x_k \in A$ and $x_l \in B$ imply $k < l$.

Each of the $n!$ permutations can separate at most one of the pairs (A_i, B_i), $i = 1, \ldots, m$. Indeed, suppose that $(x_1, x_2, \ldots, x_n)$ separates two pairs (A_i, B_i) and (A_j, B_j) with $i \neq j$, and assume that $\max\{k \mid x_k \in A_i\} \leq \max\{k \mid x_k \in A_j\}$. Since the permutation separates the pair (A_j, B_j),

$$\min\{l \mid x_l \in B_j\} > \max\{k \mid x_k \in A_j\} \geq \max\{k \mid x_k \in A_i\}$$

which implies that $A_i \cap B_j = \emptyset$, contradicting the assumption.

We now estimate the number of permutations separating one fixed pair. If $|A|{=}a$ and $|B| = b$ and A and B are disjoint then the pair (A, B) is separated by exactly

$$\binom{n}{a+b} a!b!(n-a-b)! = n!\binom{a+b}{a}^{-1}$$

permutations. Here $\binom{n}{a+b}$ counts the number of choices for the positions of $A \cup B$ in the permutation; having chosen these positions, A has to occupy the first a places, giving $a!$ choices for the order of A, and $b!$ choices for the order of B; the remaining elements can be chosen in $(n-a-b)!$ ways.

Since no permutation can separate two different pairs (A_i, B_i), summing up over all m pairs we get all permutations at most once

$$\sum_{i=1}^{m} n!\binom{a_i + b_i}{a_i}^{-1} \leq n!$$

and the desired bound (9.1) follows. □

Theorem 9.11. (Erdős–Hajnal–Moon 1964) *Every τ-critical graph H has at most $\binom{\tau(H)+1}{2}$ edges.*

Proof. Let H be a τ-critical graph with $\tau(H) = t$, and let $E = \{e_1, \ldots, e_m\}$ be the edges of H. Since H is critical, $E \setminus \{e_i\}$ has a $(t-1)$-element vertex-cover S_i. Then $e_i \cap S_i = \emptyset$ while $e_j \cap S_i \neq \emptyset$, if $j \neq i$. We can therefore apply Theorem 9.10 and obtain that $m \leq \binom{2+(t-1)}{2} = \binom{t+1}{2}$, as desired. □

Proof of Theorem 9.7

We consider graphs on vertex-set $[n] = \{1, \ldots, n\}$. We have a set X of $\binom{n}{2}$ boolean variables x_e corresponding to edges. Each graph $G = ([n], E)$ is specified by setting the values 0 and 1 to these variables: $E = \{e \mid x_e = 1\}$. The function $\mathrm{VC}(n, k)$ accepts G iff $\tau(G) \geq k + 1$.

Let $\mathrm{Crit}(n,k)$ denote the set of all τ-critical graphs on $[n] = \{1,\ldots,n\}$ with $\tau(H) = k+1$. Observe that graphs in $\mathrm{Crit}(n,k)$ are exactly the minterms of $\mathrm{VC}(n,k)$, that is, the smallest with respect to the number of edges graphs accepted by $\mathrm{VC}(n,k)$.

Given a family F of functions $f : [n] \to [r]$, let $\Phi_F(X)$ be the OR over all graphs $H \in \mathrm{Crit}(r,k)$ and all functions $f \in F$ of the following monotone formulas

$$K_{f,H}(X) = \bigwedge_{\{a,b\}\in E(H)} \bigvee_{e\in f^{-1}(a)\times f^{-1}(b)} x_e.$$

The formula Φ_F accepts a given graph $G = ([n], E)$ iff there exists a graph $H \in \mathrm{Crit}(r,k)$ and a function $f \in F$ such that for each edge $\{a,b\}$ of H there is at least one edge in G between $f^{-1}(a)$ and $f^{-1}(b)$.

A family F of functions $f : [n] \to [r]$ is *s-perfect* if for every subset $S \subseteq [n]$ of size $|S| = s$ there is an $f \in F$ such that $|f(S)| = |S|$. That is, for every s-element subset of $[n]$ at least one function in F is one-to-one when restricted to this subset. Such families are also known in the literature as (n,r,s)-perfect hash families.

Lemma 9.12. *If F is an (n,r,s)-perfect hash family with $s = 2(k+1)$ and $r \geq s$, then the formula Φ_F computes $\mathrm{VC}(n,k)$.*

Proof. Since the formula is monotone, it is enough to show that:

(a) $\tau(G) \geq k+1$ for every graph G accepted by Φ_F, and
(b) Φ_F accepts all graphs from $\mathrm{Crit}(n,k)$.

To show (a), suppose that Φ_F accepts some graph G. Then this graph must be accepted by some sub-formula $K_{f,H}$ with $f \in F$ and $H \in \mathrm{Crit}(r,k)$. That is, for every edge $\{a,b\}$ in H there must be an edge in G joining some vertex $i \in f^{-1}(a)$ with some vertex $j \in f^{-1}(b)$. Hence, if a set S covers the edge $\{i,j\}$, that is, if $S \cap \{i,j\} \neq \emptyset$, then the set $f(S)$ must cover the edge $\{a,b\}$. This means that, for any vertex-cover S in G, the set $f(S)$ is a vertex-cover in H. Taking a minimal vertex-cover S in G we obtain $\tau(G) = |S| \geq |f(S)| \geq \tau(H) = k+1$.

To show (b), take an arbitrary graph $G = ([n], E)$ in $\mathrm{Crit}(n,k)$, and let U be the set of its non-isolated vertices. By Corollary 9.9, $|U| \leq 2\tau(G) = 2(k+1) \leq s$. By the definition of F, some function $f : [n] \to [r]$ must be one-to-one on U. For $i,j \in U$ join $a = f(i)$ and $b = f(j)$ by an edge iff $\{i,j\} \in E$. Since $G \in \mathrm{Crit}(n,k)$ and f is one-to-one on all non-isolated vertices of G, the resulting graph H belongs to $\mathrm{Crit}(r,k)$. Moreover, for every edge $\{a,b\}$ of H, the pair $e = \{i,j\}$ with $f(i) = a$ and $f(j) = b$ is an edge of G, implying that $x_e = 1$. This means that the sub-formula $K_{f,H}$ of Φ_F, and hence, the formula Φ_F itself must accept G. □

Let us now estimate the number of gates in the formula Φ_F. Using a simple counting argument, Mehlhorn and Schmidt (1982) shows that (n,r,s)-perfect hash families F of size $|F| \leq se^{s^2/r}\log n$ exist for all $2 \leq s \leq r \leq n$. In our case we can take $r = s = 2(k+1)$. Hence, $|F| = \mathcal{O}(\log n)$ for every constant k.

If we allow unbounded fanin, then each sub-formula $K_{f,H}$ contributes just one AND gate. Hence, Φ_F has at most $|\mathrm{Crit}(r,k)| + |F|$ unbounded-fanin AND gates. The fanin of each AND gate is actually bounded by the number of edges in the corresponding graph $H \in \mathrm{Crit}(r,k)$ which, by Theorem 9.11, does not exceed $l := \binom{k+2}{2} = \mathcal{O}(1)$. Hence, $|\mathrm{Crit}(r,k)|$ does not exceed $\binom{r^2}{l} = \mathcal{O}(1)$. Thus, for every constant k, we have only $\mathcal{O}(|F|) = \mathcal{O}(\log n)$ fanin-2 AND gates in Φ_F. Each of these gates takes at most $\mathcal{O}(n^2)$ fanin-2 OR gates as inputs. Thus, the total size of our formula Φ_F is $\mathcal{O}(n^2 \log n)$, as desired. For growing k, the upper bound has the form $\mathcal{O}(kC^{k^2} n^2 \log n)$ for a constant C, which is polynomial as long as $k = \mathcal{O}(\sqrt{\log n})$.

We thus constructed a monotone formula Φ_F for the vertex cover function $\mathrm{VC}(n,k)$. Since this function is the dual function of the clique function $\mathrm{CLIQUE}(n, n-k)$, we can just replace OR gates by AND gates (and vice versa) in this formula to obtain a monotone formula for $\mathrm{CLIQUE}(n, n-k)$. This completes the proof of Theorem 9.7. □

Remark 9.13. Observe that the formula Φ_F for $\mathrm{VC}(n,k)$ is multilinear, that is, inputs to each its AND gate are computed from disjoint sets of variables. On the other hand, Krieger (2007) shows that *every* monotone multilinear circuit for the dual function $\mathrm{CLIQUE}(n, n-k)$ requires at least $\binom{n}{k}$ gates. This gives an example of a boolean function, whose dual requires much larger multilinear circuits than the function itself.

Remark 9.14. Using *explicit* perfect hash families we can obtain explicit circuits. For fixed values of r and s, infinite classes of (n,r,s)-perfect hash families F of size $|F| = O(\log n)$ were constructed by Wang and Xang (2001) using algebraic curves over finite fields. With this construction Theorem 9.7 gives explicit monotone formulas.

The construction in Wang and Xang (2001) is almost optimal: the family has only a *logarithmic* in n number of functions. The construction is somewhat involved. On the other hand, perfect hash families of *poly-logarithmic* size can be constructed very easily.

Let $s \geq 1$ be a fixed integer and $r = 2^s$. Let $M = \{m_{a,i}\}$ be an $n \times b$ matrix with $b = \lceil \log n \rceil$ columns whose rows are distinct 0-1 vectors of length b. Let $h_1, \ldots, h_b$ be the family of functions $h_i : [n] \to \{0,1\}$ determined by the columns of M; hence, $h_i(a) = m_{a,i}$. Let also $g : \{0,1\}^s \to [r]$ be defined by $g(x) = \sum_{i=1}^{s} x_i 2^{i-1}$.

By Bondy's theorem (Bondy 1972), the projections of any set of $s+1$ distinct binary vectors on some set of s coordinates must all be distinct. Hence, for any set $a_1, \ldots, a_{s+1}$ of $s+1$ rows there exist s columns $h_{i_1}, \ldots, h_{i_s}$ such that all $s+1$ vectors $(h_{i_1}(a_j), \ldots, h_{i_s}(a_j))$, $j = 1, \ldots, s+1$ are distinct. Therefore, the function $f(x) = g(h_{i_1}(x), \ldots, h_{i_s}(x))$ takes different values on all $s+1$ points $a_1, \ldots, a_{s+1}$. Thus, taking the superposition of g with $\binom{b}{s} \leq \log^s n$ s-tuples of functions $h_1, \ldots, h_b$, we obtain a family F of $|F| \leq \log^s n$ functions $f : [n] \to [r]$ which is $(s+1)$-perfect.

9.3 The Monotone Switching Lemma

In Razborov's method of approximations one only uses DNFs to approximate gates. In this way, OR gates can be easily approximated: an OR of DNFs is a DNF, and we only need to keep its small enough. The case of AND gates is, however, more complicated. So, a natural idea to try to approximate by *both* DNFs and CNFs. When appropriately realized, this idea leads to a general, and relatively simple lower-bounds criterion for monotone circuits. Due to the symmetry between DNFs and CNFs, this criterion is often much easier to apply and yields exponential lower bounds for many functions, including the clique function.

> Still, there are functions—like the perfect matching function—for which the criterion seems to fail. This is why we will discuss Razborov's method later in Sect. 9.10 in full detail: unlike the general criterion, which we are going to present now, Razborov's method is much more subtle, tailor made for the *specific* function one deals with and can be applied in situations where the general criterion fails to produce strong lower bounds. Yet another reason to include Razborov's proof for the perfect matching function is that this function belongs to P, and the proof was never treated in a book.

Our goal is to show that, if a monotone boolean function can be computed by a small monotone circuit, then it can be approximated by small monotone CNFs and DNFs. Thus, in order to prove that a function requires large circuits it is enough to show that it does not have a small CNF/DNF approximation. The proof of this will be based on the "monotone switching lemma" allowing us to switch between CNFs and DNFs, and vice versa.

By a monotone k-CNF (conjunctive normal form) we will mean an And of an arbitrary number of monotone clauses, each being an Or of at most k variables. Dually, a monotone k-DNF is an Or of an arbitrary number of monomials, each being an And of at most k variables. In an *exact* k-CNF all clauses must have *exactly* k distinct variables; *exact* k-DNFs are defined similarly. For two boolean functions f and g of the same set of variables, we write $f \leq g$ if $f(x) \leq g(x)$ for all input vectors x. For a CNF/DNF C we will denote by $|C|$ the number of clauses/monomials in it.

The following lemma was first proved in Jukna (1999) in terms of so-called "finite limits", a notion suggested by Sipser (1985); we will also use this notion later (in Sect. 11.3) to prove lower bounds for depth-3 circuits. In terms of DNFs and CNFs the lemma was then proved by Berg and Ulfberg (1999). Later, a similar lemma was used by Harnik and Raz (2000) to improve the numerically strongest known lower bound $2^{\Omega(n^{1/3}/\log n)}$ of Andreev (1987b) to $2^{\Omega((n/\log n)^{1/3})}$. The idea of the lemma itself was also implicit in the work of Haken (1995).

Lemma 9.15. (Monotone Switching Lemma) *For every* $(s-1)$*-CNF* $f_{\rm cnf}$ *there is an* $(r-1)$*-DNF* $f_{\rm dnf}$ *and an exact* r*-DNF* D *such that*

$$f_{\rm dnf} \leq f_{\rm cnf} \leq f_{\rm dnf} \vee D \quad \text{and} \quad |D| \leq (s-1)^r \,. \tag{9.2}$$

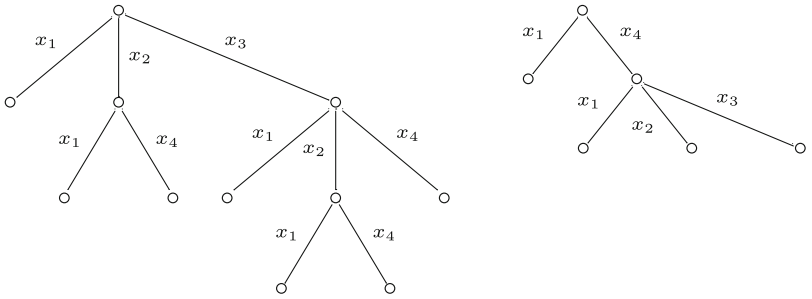

Fig. 9.2 Two DNF-trees of the same 3-CNF $f_{\text{cnf}} = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_1 \vee x_4)$. The second tree is obtained by parsing the clauses of f_{cnf} in the reverse order

Dually, for every $(r-1)$*-DNF* f_{dnf} *there is an* $(s-1)$*-CNF* f_{cnf} *and an exact* s*-CNF* C *such that*

$$f_{\text{cnf}} \wedge C \leq f_{\text{dnf}} \leq f_{\text{cnf}} \quad \textit{and} \quad |C| \leq (r-1)^s \,. \tag{9.3}$$

Proof. We prove the first claim (the second is dual). Let $f_{\text{cnf}} = q_1 \wedge \cdots \wedge q_l$ be an $(s-1)$-CNF; hence, each clause q_i has $|q_i| \leq s-1$ variables. It will be convenient to identify clauses and monomials with the *sets* of indices of their variables. We say that a monomial p *pierces* a clause q_i if $p \cap q_i \neq \emptyset$.

We associate with f_{cnf} the following "transversal" tree T of fan-out at most $s-1$ (see Fig. 9.2).

The first node of T corresponds to the first clause q_1, and the outgoing $|q_1|$ edges are labeled by the variables from q_1. Suppose we have reached a node v, and let p be the monomial consisting of the labels of edges from the root to v. If p pierces all the clauses of f_{cnf}, then v is a leaf. Otherwise, let q_i be the *first* clause such that $p \cap q_i = \emptyset$. Then the node v has $|q_i|$ outgoing edges labeled by the variables in q_i.

Note that the resulting tree T depends on what ordering of clauses of f_{cnf} we fix, that is, in which order we parse the clauses (see Fig. 9.2). Still, for any such tree we have that, for every assignment $x \in \{0,1\}^n$, $f_{\text{cnf}}(x) = 1$ if and only if x is consistent with at least one path from the root to a leaf of T. This holds because $f_{\text{cnf}}(x) = 1$ iff the set $S_x = \{i \mid x_i = 1\}$ intersects all clauses $q_1, \ldots, q_l$.

Some paths in T may be longer than $r-1$. So, we now cut off these long paths. Namely, let f_{dnf} be the OR of all paths of length at most $r-1$ ending in leafs, and D be the OR of all paths of length exactly r. Observe that for every assignment $x \in \{0,1\}^n$:

- $f_{\text{dnf}}(x) = 1$ implies $f_{\text{cnf}}(x) = 1$, and
- $f_{\text{cnf}}(x) = 1$ implies $f_{\text{dnf}}(x) = 1$ or $D(x) = 1$.

Thus, $f_{\text{dnf}} \leq f_{\text{cnf}} \leq f_{\text{dnf}} \vee D$. Finally, we also have that $|D| \leq (s-1)^r$, because every node of T has fan-out at most $s-1$. □

Most important in the Switching Lemma is that the exact DNFs and CNFs correcting possible errors contain only $(s-1)^r$ monomials instead of all $\binom{n}{r}$ possible monomials, and only $(r-1)^s$ clauses instead of all $\binom{n}{s}$ possible clauses.

9.4 The Lower-Bounds Criterion

We now give a general lower-bounds criterion for monotone circuits.

Definition 9.16. Let f be a monotone boolean function of n variables. We say that f is *t-simple* if for every pair of integers $2 \le r, s \le n$ there exists an exact s-CNF C, an exact r-DNF D, and a subset $I \subseteq [n]$ of size $|I| \le s-1$ such that

(a) $|C| \le t \cdot (r-1)^s$ and $|D| \le t \cdot (s-1)^r$, and
(b) Either $C \le f$ or $f \le D \vee \bigvee_{i \in I} x_i$ (or both) hold.

Theorem 9.17. *If a monotone boolean function can be computed by a monotone circuit of size t, then f is t-simple.*

Proof. Let $F(x_1, \ldots, x_n)$ be a monotone boolean function, and suppose that F can be computed by a monotone circuit of size t. Our goal is to show that the function F is t-simple. To do this, fix an arbitrary pair of integer parameters $2 \le s, r \le n$.

Let $f = g * h$ be a gate in our circuit. That is, f is a function computed at some node of the circuit, and g and h are functions computed at its inputs. By an *approximator* of this gate we will mean a pair $(f_{\text{cnf}}, f_{\text{dnf}})$, where f_{cnf} is an $(s-1)$-CNF (a *left* approximator of f) and f_{dnf} is an $(r-1)$-DNF (a *right* approximator of f) such that $f_{\text{dnf}} \le f_{\text{cnf}}$.

We say that such an approximator $f_{\text{cnf}}, f_{\text{dnf}}$ of f introduces a new error on input $x \in \{0,1\}^n$ if the approximators of g and of h did not make an error on x, but the approximator of f does. That is, $g_{\text{cnf}}(x) = g_{\text{dnf}}(x) = g(x)$ and $h_{\text{cnf}}(x) = h_{\text{dnf}}(x) = h(x)$, but either $f_{\text{cnf}}(x) \ne f(x)$ or $f_{\text{dnf}}(x) \ne f(x)$.

We define approximators inductively as follows.

Case 1: f is an input variable, say, $f = x_i$. In this case we take $f_{\text{cnf}} = f_{\text{dnf}} := x_i$. It is clear that this approximator introduces no errors.

Case 2: f is an And gate, $f = g \wedge h$. In this case we take $f_{\text{cnf}} := g_{\text{cnf}} \wedge h_{\text{cnf}}$ as the left approximator of f; hence, f_{cnf} introduces no new errors. To define the right approximator of f we use Lemma 9.15 to convert f_{cnf} into an $(r-1)$-DNF f_{dnf}; hence, $f_{\text{dnf}} \le f_{\text{cnf}}$. Let E_f be the set of inputs on which f_{dnf} introduces a new error, that is,

$$E_f := \{x \mid f(x) = f_{\text{cnf}}(x) = 1 \text{ but } f_{\text{dnf}}(x) = 0\}.$$

By Lemma 9.15, all these errors can be "corrected" by adding a relatively small exact r-DNF: there is an exact r-DNF D such that $|D| \le (s-1)^r$ and $D(x) = 1$ for all $x \in E_f$.

Case 3: f is an Or gate, $f = g \vee h$. This case is dual to Case 2. We take $f_{\text{dnf}} := g_{\text{dnf}} \vee h_{\text{dnf}}$ as the right approximator of f; hence, f_{dnf} introduces no new errors. To define the left approximator of f we use Lemma 9.15 to convert f_{dnf} into an $(s-1)$-CNF f_{cnf}; hence, $f_{\text{dnf}} \le f_{\text{cnf}}$. Let E_f be the set of inputs on which f_{cnf} introduces a new error, that is,

$$E_f := \{x \mid f(x) = f_{\text{dnf}}(x) = 0 \text{ but } f_{\text{cnf}}(x) = 1\}.$$

By Lemma 9.15, all these errors can be "corrected" by adding a relatively small exact s-CNF: there is an exact s-CNF C such that $|C| \le (r-1)^s$ and $C(x) = 0$ for all $x \in E_f$.

Proceeding in this way we will reach the last gate of our circuit computing the given function F. Let $(F_{\rm cnf}, F_{\rm dnf})$ be its approximator, and let E be the set of all inputs $x \in \{0,1\}^n$ on which F differs from at least one of the functions $F_{\rm cnf}$ or $F_{\rm dnf}$. Since at input gates (= variables) no error was made, for every such input $x \in E$, the corresponding error must be introduced at some intermediate gate. That is, for every $x \in E$ there is a gate f such that $x \in E_f$ (approximator of f introduces an error on x for the first time). But we have shown that, for each gate, all these errors can be corrected by adding an exact s-CNF of size at most $(r-1)^s$ or an exact r-DNF of size at most $(s-1)^r$. Since we have only t gates, all such errors $x \in E$ can be corrected by adding an exact s-CNF C of size at most $t \cdot (r-1)^s$ and an exact r-DNF D of size at most $t \cdot (s-1)^r$, that is, for all inputs $x \in \{0,1\}^n$, we have

$$C(x) \wedge F_{\rm cnf}(x) \le F(x) \le F_{\rm dnf}(x) \vee D(x)\,,$$

where $F_{\rm dnf} \le F_{\rm cnf}$. This already implies that the function F is t-simple. Indeed, if the CNF $F_{\rm cnf}$ is empty (that is, if $F_{\rm cnf} \equiv 1$) then $C \le F$, and we are done. Otherwise, $F_{\rm cnf}$ must contain some clause q of length at most $s-1$, say, $q = \bigvee_{i\in I} x_i$ for some $I \subseteq [n]$ of size $|I| \le s-1$. Since clearly $F_{\rm cnf} \le q$, the condition $F_{\rm dnf} \le F_{\rm cnf}$ implies $F \le F_{\rm dnf} \vee D \le F_{\rm cnf} \vee D \le q \vee D$, as desired. This completes the proof of Theorem 9.17. □

In applications, boolean functions f are usually defined as set-theoretic predicates. In this case we say that f accepts a set $S \subseteq \{1,\ldots,n\}$ and write $f(S) = 1$ if and only if f accepts its incidence vector. Let $\overline{S} = \{1,\ldots,n\} \setminus S$ denote the complement of S. We say that a set S is a

- *Positive input* for f if $f(S) = 1$;
- *Negative input* for f if $f(\overline{S}) = 0$.

Put differently, a positive (negative) input is a set of variables which, if assigned the value 1 (0), forces the function to take the value 1 (0) regardless of the values assigned to the remaining variables. The minimal (under set inclusion) positive inputs for f are called *minterms* of f. Similarly, the maximal negative inputs for f are called *maxterms* of f.

Note that one and the same set S can be both a positive and a negative input! For example, if $f(x_1,x_2,x_3)$ outputs 1 iff $x_1 + x_2 + x_3 \ge 2$, then $S = \{1,2\}$ is both positive and negative input for f, because $f(1,1,x_3) = 1$ and $f(0,0,x_3) = 0$.

To re-formulate the definition of t-simplicity (Definition 9.16) in terms of positive/negative inputs, note that if C is a CNF, then $C \le f$ means that every negative input of f must contain at least one clause of C (looked at as set of indices of its variables). Similarly, $f \le D \vee \bigvee_{i\in I} x_i$ means that every positive input must either intersect the set I or contain at least one monomial of D. Thus, if $\mathcal{F}_1$ ($\mathcal{F}_0$) is a family of positive (negative) inputs of f, and $\#_k(\mathcal{F})$ denotes the maximum number

of members of $\mathcal{F}$ containing a fixed k-element set, then Theorem 9.17 gives the following more explicit lower bound.

Theorem 9.18. *For every integers $2 \le r, s \le n$, every monotone circuit computing f must have size at least the minimum of*

$$\frac{|\mathcal{F}_1| - (s-1)\cdot \#_1(\mathcal{F}_1)}{(s-1)^r \cdot \#_r(\mathcal{F}_1)} \quad \textit{and} \quad \frac{|\mathcal{F}_0|}{(r-1)^s \cdot \#_s(\mathcal{F}_0)}\,.$$

That is, a monotone boolean function requires large monotone circuits if its positive as well as negative inputs are "scattered" well enough.

9.5 Explicit Lower Bounds

In order to show that a given boolean function cannot be computed by a monotone circuit of size at most t, it is enough, by Theorem 9.17, to show that the function is not t-simple for at least one(!) choice of parameters s and r. In this section we demonstrate how this can be used to derive strong lower bounds for concrete boolean functions.

9.5.1 Detecting Triangles

We begin with the simplest example, yielding a polynomial lower bound. We will also present more "respectable" applications leading to exponential lower bounds, but this special case already demonstrates the common way of reasoning fairly well.

Let us consider a monotone boolean function Δ_n, whose input is an undirected graph on n vertices, represented by $v = \binom{n}{2}$ variables, one for each possible edge. The value of the function is 1 if and only if the graph contains a triangle (three incident vertices). Clearly, there is a monotone circuit of size $\mathcal{O}(n^3)$ computing this function: just test whether any of $\binom{n}{3}$ triangles is present in the graph. Thus, the following theorem is tight, up to a poly-logarithmic factor.

Theorem 9.19. *Any monotone circuit, detecting whether a given n-vertex graph is triangle-free, must have $\Omega(n^3/\log^4 n)$ gates.*

Proof. Let t be the minimal number for which Δ_n is t-simple. By Theorem 9.17, it is enough to show that $t = \Omega(n^3/\log^4 n)$. For this proof, we take $s := \lfloor 5\log^2 n \rfloor$ and $r := 2$. According to the definition of t-simplicity, we have only two possibilities.

Case 1: Every positive input for Δ_n either intersects a fixed set I of s edges, or contains at least one of $L \le ts^r = ts^2$ 2-element sets of edges $R_1, \ldots, R_L$.

As positive inputs for Δ_n we take all triangles, that is, graphs on n vertices with exactly one triangle; we have $\binom{n}{3}$ such graphs. At most $s(n-2)$ of them will have

an edge in I. Each of the remaining triangles must contain one of ts^2 given pairs of edges R_i. Since two edges can lie in at most one triangle, we conclude that, in this case,

$$t \geq \frac{\binom{n}{3} - s(n-2)}{s^2} = \Omega(n^3/\log^4 n)\,.$$

Case 2: Every negative input for Δ_n contains at least one of $t(r-1)^s = t$ sets of edges $S_1, \ldots, S_t$, each of size $|S_i| = s$.

In this case we consider the graphs $E = E_1 \cup E_2$ consisting of two disjoint non-empty cliques E_1 and E_2 (we consider graphs as sets of their edges). Each such graph E is a negative input for Δ_n, because its complement is a bipartite graph, and hence, has no triangles. The number of such graphs is a half of the number 2^n of all binary strings of length n excluding the all-0 and all1 strings. Hence, we have $2^{n-1} - 1$ such graphs, and each of them must contain at least one of the sets $S_1, \ldots, S_t$. Every of these sets of edges S_i is incident to at least $\sqrt{2s}$ vertices, and if $E \supseteq S_i$ then all these vertices must belong to one of the cliques E_1 or E_2. Thus, at most $2^{n-\sqrt{2s}} - 1$ of our negative inputs E can contain one fixed set S_i, implying that, in this case,

$$t \geq \frac{2^{n-1}-1}{2^{n-\sqrt{2s}}-1} \geq 2^{\sqrt{2s}-1} \geq 2^{3\log n} \geq n^3\,.$$

Thus, in both cases, $t = \Omega(n^3/\log^4 n)$, and we are done. □

9.5.2 Graphs of Polynomials

Our next example is the following monotone boolean function introduced by Andreev (1985). Let $q \geq 2$ be a prime power, and set $d := \lfloor (q/\ln q)^{1/2}/2 \rfloor$. Consider boolean $q \times q$ matrices $A = (a_{i,j})$. Given such a matrix A, we are interested in whether it contains a graph of a polynomial $h : \mathrm{GF}(q) \to \mathrm{GF}(q)$, that is, whether $a_{i,h(i)} = 1$ for all rows $i \in \mathrm{GF}(q)$.

Let f_n be a monotone boolean function of $n = q^2$ variables such that $f_n(A) = 1$ iff A contains a graph of at least one polynomial over $\mathrm{GF}(q)$ of degree at most $d-1$. That is,

$$f_n(X) = \bigvee_h \bigwedge_{i \in \mathrm{GF}(q)} x_{i,h(i)}\,,$$

where h ranges over all polynomials over $\mathrm{GF}(q)$ of degree at most $d-1$. Since we have at most q^d such polynomials, the function f_n can be computed by a monotone boolean circuit of size at most q^{d+1}, which is at most $n^{\mathcal{O}(d)} = 2^{\mathcal{O}(n^{1/4}\sqrt{\ln n})}$. We will now show that this trivial upper bound is almost optimal.

Theorem 9.20. *Any monotone circuit computing the function f_n has size at least $2^{\Omega(n^{1/4}\sqrt{\ln n})}$.*

Proof. Take a minimal t for which the function f_n is t-simple. Since $n = q^2$ and (by our choice) $d = \Theta(n^{1/4}\sqrt{\ln n})$, it is enough by Theorem 9.17 to show that $t \geq q^{\Omega(d)}$. For this proof, we take $s := \lfloor d \ln q \rfloor$ and $r := d$, and consider input matrices as bipartite $q \times q$ graphs. In the proof we will use the well-known fact that no two distinct polynomials of degree at most $d-1$ can coincide on d points. According to the definition of t-simplicity, we have only two possibilities.

Case 1: Every positive input for f_n either intersects a fixed set I of at most s edges, or contains at least one of $L \leq ts^r$ r-element sets of edges $R_1, \ldots, R_L$.

Graphs of polynomials of degree at most $d-1$ are positive inputs for f_n. Each set of l $(1 \leq l \leq d)$ edges is contained in either 0 or precisely q^{d-l} of such graphs. Hence, at most sq^{d-1} of these graphs can contain an edge in I, and at most q^{d-r} of them can contain any of the given graphs R_i. Therefore, in this case we have

$$t \geq \frac{q^d - sq^{d-1}}{s^r \cdot q^{d-r}} = \left(\frac{q}{s}\right)^{\Omega(r)} = q^{\Omega(d)}.$$

Case 2: Every negative input for f_n contains at least one of $K \leq tr^s$ s-element sets of edges $S_1, \ldots, S_K$.

Let E be a random bipartite graph, with each edge appearing in E independently with probability $\gamma := (2d \ln q)/q$. Since there are only q^d polynomials of degree at most $d-1$, the probability that the complement of E will contain the graph of at least one of them does not exceed $q^d(1-\gamma)^q \leq q^{-d}$, by our choice of γ. Hence, with probability at least $1 - q^{-d}$, the graph E is a negative input for f. On the other hand, each of the sets S_i is contained in E with probability $\gamma^{|S_i|} = \gamma^s$. Thus, in this case,

$$t \geq \frac{1 - q^{-d}}{r^s \gamma^s} = \left(\frac{q}{2d^2 \ln q}\right)^{\Omega(s)} = 2^{\Omega(s)} = q^{\Omega(d)},$$

where the third inequality holds for all $d \leq (q/\ln q)^{1/2}/2$.

We have proved that the function f can be t-simple only if $t \geq q^{\Omega(d)}$. By Theorem 9.17, this function cannot be computed by monotone circuits of size smaller than $q^{\Omega(d)}$. □

9.6 Circuits with Real-Valued Gates

We now consider monotone circuits where, besides boolean AND and OR gates, one may use arbitrary monotone *real-valued* functions $\varphi : \mathbb{R}^2 \to \mathbb{R}$ as gates. Such a function φ is *monotone* if $\varphi(x_1, x_2) \leq \varphi(y_1, y_2)$ whenever $x_1 \leq y_1$ and $x_2 \leq y_2$. The corresponding circuits are called *monotone real circuit*.

First lower bounds for monotone circuits with real-valued gates were proved by Pudlák ~~et al.~~ (1997), via an extension of Razborov's argument, and by

Haken and Cook (1999), via an extension of the "bottleneck counting" argument of Haken (1995).

As in boolean circuits, inputs for such circuits are also binary strings $x \in \{0,1\}^n$; the output must also be a binary bit 0 or 1. But at each intermediate gate any monotone function $f : \{0,1\}^n \to \mathbb{R}$ may be computed. Hence, unlike in boolean case, here we have uncountable number of possible gates $\varphi : \mathbb{R}^2 \to \mathbb{R}$, and one may expect that at least some monotone boolean functions can be computed much more efficiently by such circuits. Exercise 9.6 shows that this intuition is correct: so-called "slice functions" can be computed by a very small monotone circuit with real-valued gates, but easy counting shows that most slice functions cannot be computed by boolean circuits of polynomial size, even if NOT gates are allowed! Thus, monotone real circuits may be even exponentially more powerful than circuits over $\{\wedge, \vee, \neg\}$.

It is therefore somewhat surprising that the (simple) criterion for boolean circuits (Theorem 9.17) remains true also for circuits with real-valued gates. The only difference from the boolean case is that now in the definition of t-simplicity we take slightly larger CNFs and DNFs, which does not greatly change the asymptotic values of the resulting lower bounds.

We say that a monotone boolean function f is *weakly t-simple* if the conditions in Definition 9.16 hold with (a) replaced by

(a′) $|C| \le t \cdot (2r)^{s+1}$ and $|D| \le t \cdot (2s)^{r+1}$

That is, the only difference from the definition of t-simplicity is a slightly larger upper bound on the number of clauses in C and monomials in D.

Theorem 9.21. (Criterion for Real Circuits) *Let f be a monotone boolean function. If f can be computed by a monotone real circuit of size t then f is weakly t-simple.*

Proof. The theorem was first proved in (Jukna 1999) using finite limits. A much simpler proof, which we present below, is due to Avi Wigderson. The argument is similar to that in the boolean case (Theorem 9.21). We only have to show how to construct the approximators for real-valued gates. The idea is to consider *thresholds* of real gates and approximate the thresholded values. For a real-valued function $f : \{0,1\}^n \to \mathbb{R}$ and a real number a, let $f^{(a)}$ denote the boolean function that outputs 1 if $f(x) \ge a$, and outputs 0 otherwise.

Now let $\varphi : \mathbb{R}^2 \to \mathbb{R}$ be a gate at which the function $f(x)$ is computed, and let $g(x)$ and $h(x)$ be functions $g, h : \{0,1\}^n \to \mathbb{R}$ computed at the inputs of this gate. A simple (but crucial) observation is that then

$$\varphi\big(g(x), h(x)\big) \ge a \quad \Longleftrightarrow \quad \exists b, c : \; g(x) \ge b, \;\; h(x) \ge c \;\text{ and }\; \varphi(b,c) \ge a\,.$$

The ($\Rightarrow$) direction is trivial: just take $b = g(x)$ and $c = h(x)$. The other direction ($\Leftarrow$) follows from the monotonicity of φ: $\varphi(g(x), h(x)) \ge \varphi(b,c) \ge a$.

Together with the fact that $f^{(a)}(x) = 1$ iff $\varphi(g(x), h(x)) \ge a$, this allows us to express each threshold function $f^{(a)}$ of a gate $f = \varphi(g,h)$ from the thresholds of its input gates as:

$$f^{(a)} = \bigvee_{\varphi(b,c)\geq a} (g^{(b)} \wedge h^{(c)}) \tag{9.4}$$

as well as

$$f^{(a)} = \bigwedge_{\varphi(b,c)<a} (g^{(b)} \vee h^{(c)}) \,. \tag{9.5}$$

It is convenient to think these expressions as an infinite AND and an infinite OR, respectively. However, since the number of settings $x \in \{0,1\}^n$ for input variables is finite, the real gates take only finite number of possible values, and we therefore only need finite expressions.

Fix a pair $1 \leq s, r < n$ of integer parameters. As before, every threshold $f^{(a)}$ is approximated by two functions: an s-CNF $f^{(a)}_{\text{cnf}}$ (*left* approximator) and an r-DNF $f^{(a)}_{\text{dnf}}$ (*right* approximator). The approximators for the thresholds of the input variables are 0, 1, or the variable itself, depending on the value of the threshold; they can always be represented by at most one literal and thus never fail.

Now let $f = \varphi(g,h)$ be an intermediate gate with two input gates g and h, and suppose that, for all (finitely many!) reals b, c, the left and right approximators for threshold functions $g^{(b)}$ and $h^{(c)}$ of its input gates are already constructed.

To construct the *left* approximator $f^{(a)}_{\text{cnf}}$ of $f^{(a)}$ from the approximators of its two input gates g and h, we first consider the representation

$$f^{(a)} = \bigvee_{\varphi(b,c)\geq a} (g^{(b)}_{\text{dnf}} \wedge h^{(c)}_{\text{dnf}}) \,.$$

Since the monomials in the r-DNFs $g^{(b)}_{\text{dnf}}$ and $h^{(c)}_{\text{dnf}}$ have length at most r, all the subexpressions $g^{(b)}_{\text{dnf}} \wedge h^{(c)}_{\text{dnf}}$ can be turned into a single $2r$-DNF D_a such that

$$D_a(x) = 1 \quad \text{iff} \quad f^{(a)}(x) = 1 \quad \text{iff} \quad f(x) \geq a \,. \tag{9.6}$$

After that we use the same procedure as before (that is, Lemma 9.15) to convert this DNF into an s-CNF $f^{(a)}_{\text{cnf}}$. This can be done for each (of the finitely many) threshold values a, and we only need to ensure that the number of errors introduced when approximating the whole gate f does not depend on this number of thresholds.

When forming the s-CNF $f^{(a)}_{\text{cnf}}$, we introduce errors as we throw away clauses that become longer than s. We want to count the number of inputs $x \in \{0,1\}^n$ such that $f^{(a)}(x) = 0$ while $f^{(a)}_{\text{cnf}}(x) = 1$ for some a, that is, the union over a of the errors introduced in a gate by $f^{(a)}_{\text{cnf}}$. To do this, let us list in the increasing order $a_1 < a_2 < \ldots < a_N$ all the $N \leq 2^n$ possible values $f(x)$ the gate f can output when the input vector x ranges over $\{0,1\}^n$. Hence,

$$D := D_{a_1} \vee D_{a_2} \vee \cdots \vee D_{a_N}$$

is a $2r$-DNF, and this DNF makes no error on x, that is, $D(x) = f(x)$. By (9.6), we have that

$$D_{a_1} \geq D_{a_2} \geq \cdots \geq D_{a_N} .$$

That is, every monomial of $D_{a_{i+1}}$ contains at least one monomial of D_{a_i}. Hence, if $t(D)$ denotes the family of all transversals of D, that is, the family of all subsets of variables, each of which intersects all the monomials of D, then

$$t(D_{a_1}) \subseteq t(D_{a_2}) \subseteq \cdots \subseteq t(D_{a_N}) ,$$

implying that $t(D) = t(D_{a_N})$. This means that all the clauses (= transversals), which we throw away (because they are longer than s) when forming an s-CNF f_{cnf} from the DNF D, are precisely those clauses, which we would throw away when converting the $2r$-DNF D_{a_N} into an s-CNF. Thus, by Lemma 9.15, all the errors that may appear during the construction of the left approximator f_{cnf}, can be corrected by an exact $(s+1)$-CNF C of size $|C| \leq (2r)^{s+1}$. That is, for every input x such that $f(x) = 0$ but $f_{\text{cnf}}(x) = 1$, we have that $C(x) = 0$.

A dual argument can be used to bound the number of errors introduced when constructing the right approximator f_{dnf}. Note that we cannot use the DNF (9.6) for this purpose since D is a $2r$-DNF, not an r-DNF. But we can argue as above by using the expression (9.5) instead of (9.4). Then all the introduced errors can be corrected by an exact $(r+1)$-DNF D of size $|D| \leq (2s)^{r+1}$. The rest of the proof is the same as that of Theorem 9.17. □

Since the definitions of t-simple functions and of weakly t-simple function are almost the same, Theorem 9.21 allows us to extend lower bounds for the monotone *boolean* circuits (we proved above) to the monotone *real* circuits. For example, the same argument as in the proof of Theorem 9.20 yields

Theorem 9.22. *Any monotone real circuit computing the polynomial function f_n has size at least $2^{\Omega(n^{1/4}\sqrt{\ln n})}$.*

Lower bounds for monotone real circuits have found intriguing applications in proof complexity. In particular, Pudlák et al. (1997) used such bounds to prove the first exponential lower bound on the length of so-called "cutting plane proofs", a proof system for solving integer programming problems. We will describe this result in Chap. 19.

The extension of the lower-bounds criterion from monotone boolean circuits to monotone real circuits shows the power of the criterion. On the other hand, monotonicity is crucial here.

Proposition 9.23. *Any boolean function of n variables can be computed using $n-1$ real monotone fanin-2 gates and one non-monotone unary gate.*

Proof. For an input vector $x \in \{0,1\}^n$, let $\text{bin}(x) = \sum_{i=1}^{n} x_i 2^{i-1}$ be the number whose binary code is x. It is easy to see that $\text{bin}(x)$ can be computed by a circuit $C(x)$ using $n-1$ real fanin-2 gates of the form $g(u,v) = u + 2v$. This can be done via the recurrence:

$$\text{bin}(x) = x_1 + 2 \cdot \text{bin}(x') = g(x_1, \text{bin}(x'))\,,$$

where $x' = (x_2, \ldots, x_n)$. These gates are monotone.

Now, every boolean function f defines the unique set of numbers

$$L_f = \{\text{bin}(x) \mid f(x) = 1\}\,.$$

Hence, in order to compute f, it is enough to attach the (non-monotone) output gate testing whether $C(x) \in L_f$. □

9.7 Criterion for Graph Properties

Fix a set V of $|V| = n$ vertices, and let $\binom{n}{2}$ be the set of all potential edges $e = \{u, v\}$ with $u \neq v \in V$ on these vertices. Assign a boolean variable x_e to each potential edge e. Then every 0-1 vector x of length $\binom{n}{2}$ defines the graph $S_x = \{e \mid x_e = 1\}$; we consider graphs as sets $S \subseteq \binom{n}{2}$ of their edges. Thus, every boolean function f of $\binom{n}{2}$ variables defines some property of n-vertex graphs.

An example of a graph property is the clique function $f_n = \text{CLIQUE}(n, k)$ we have considered in Sects. 7.5 and 9.1. If applied directly, the symmetric lower-bounds criterion (Theorem 9.18) cannot yield strong lower bounds for this function. In this case, we can take as positive inputs of f_n the family $\mathcal{F}$ of all $\binom{n}{k}$ cliques on k vertices. But then we would only have that $\#_r(\mathcal{F}) \leq \binom{n-\sqrt{r}}{k-\sqrt{r}}$ because some sets S of $|S| = r$ edges may touch at most $\sqrt{r}$ vertices, with the worst case of S being a clique. Hence, the fraction

$$\frac{|\mathcal{F}|}{s^r \cdot \#_r(\mathcal{F})} \geq \frac{\binom{n}{k}}{s^r \binom{n-\sqrt{r}}{k-\sqrt{r}}} \approx \frac{n^{\sqrt{r}}}{s^r} = \left(\frac{n}{s^{\sqrt{r}}}\right)^{\sqrt{r}}$$

in this case is too small: we cannot take s and r large enough. The reason for this failure is that, so far, we only used a trivial measure of "length" for clauses and monomials–the total number of variables in them. But in the case of graph properties, variables x_e correspond to edges. Thus, clauses and monomials correspond in this case to graphs (sets of edges). Say a clause $c = \bigvee_{e \in S} x_e$ corresponds to the graph S. We therefore have more flexibility to define an appropriate notion of "length" of a monomial than just as the number of variables in it. We can, say, define the "length" of a graph S as the number $v(S)$ of vertices touched by (incident with) the edges in S, or as the number $\kappa(S)$ of connected components in S, or somehow else. It makes therefore sense to extend the lower-bounds criterion for the case of different length measures. We will now show that this can be done quite easily.

By a *legal length measure* we will mean any non-negative measure $\mu(S)$ of graphs satisfying the following conditions for some non-negative constants c, d:

$$\mu(S) \le \mu(S \cup \{e\}) \le \mu(S) + c \quad \text{and} \quad |S| \le \mu(S)^d \,.$$

Parameter c tells us how much the measure of a graph can increase when one edge is added, and d tells us how much smaller can the measure of a graph be when compared to the total number of edges in it. For simplicity of exposition, we will only consider length measures with $c = d = 2$. For arbitrary c and d the arguments are the same, although the bounds we get are slightly worse.

Note that the length measure $\mu(S) = |S|$ (the total number of edges) we have considered in the previous sections has all these properties. The measure $\mu(S) =$ the number of vertices touched by the edges in S also has these properties. If we could use $\mu(S)$ instead of $|S|$, then only at most $\binom{n-r}{k-r}$ of k-cliques would contain a fixed graph S with $\mu(S) = r$, and the fraction

$$\frac{|\mathcal{F}|}{s^r \cdot \#_r(\mathcal{F})} \ge \frac{\binom{n}{k}}{s^r \binom{n-r}{k-r}} \approx \left(\frac{n}{s}\right)^r$$

would then already be large enough. We have therefore only to show that our lower bounds criteria can be extended to the case of arbitrary legal length measures.

Now, when some length measure of graphs is fixed, we can define the notions of k-CNF and of exact k-CNF in a similar way. By a k-CNF *relative to* μ we will now mean a monotone CNF each whose clause has μ-length at most k. In an *exact* k-CNF *relative to* μ we require that all clauses have μ-length at least k; and similarly for DNFs.

It is not difficult to verify that the Monotone Switching Lemma remains true for any pair of length measures for clauses and for monomials. The only difference is that now we have slightly worse upper bounds on $|D|$ and $|C|$, namely $|D| \le s^{4r}$ and $|C| \le r^{4s}$.

Proof. Argue as in the proof of Lemma 9.15. Regardless of which length measure for clauses we use, each clause of length s will have at most s^2 variables. Construct the "transversal tree" T in the same manner. Having a length measure μ for monomials, we now define DNFs f_{dnf} and D in the same way with the words "monomial of length" replaced by "monomial of μ-length". Namely, the DNF f_{dnf} now consists of all paths of μ-length smaller than r, and the DNF D consists of all paths whose μ-length reached the threshold r for the first time, that is, D consists of all paths p such that $\mu(p) \ge r$ but $\mu(p') < r$, where p' is the path p without its last edge. Since adding one edge can only increase the measure by an additive factor 2, every monomial in D has length (not just μ-length) at most $2r$. Since every node of T has fan-out at most s^2, this gives the desired upper bound $|D| \le (s^2)^{2r} = s^{4r}$ on the total number of monomials in D. □

Thus, in the case of graph properties f we have a more flexible lower-bounds criterion allowing us to choose different length measures for positive inputs (graphs accepted by f) and negative inputs (graphs whose complements are rejected by f). Let η be some length measure for negative inputs, and μ be some length measure for positive inputs.

Definition 9.24. (Approximators) By an (r, s)-*approximator* of f of size t we will mean a triple $(\mathcal{R}, \mathcal{S}, I)$ where [1]

- I is a graph of η-length $\leq s$;
- $\mathcal{R}$ is a family of $|\mathcal{R}| \leq t(2s)^{4r}$ graphs of μ-length $\geq r$, and
- $\mathcal{S}$ is a family of $|\mathcal{S}| \leq t(2r)^{4s}$ graphs of η-length $\geq s$

such that at least one of the following two conditions holds:

1. Every positive input of f either intersects the graph I or contains at least one of the graphs in $\mathcal{R}$.
2. Every negative input of f contains at least one of the graphs in $\mathcal{S}$.

Theorem 9.25. *If a monotone boolean function can be computed by a monotone real circuit of size t, then it has an (r, s)-approximator of size t for any $1 \leq r, s \leq n-1$ and for every pair of length measures.*

The proof of this theorem is the same as that of Theorem 9.21: just use the modified version of the Monotone Switching Lemma. We leave a detailed proof as an exercise.

9.8 Clique-Like Problems

We consider graphs on a fixed set V of $|V| = n$ vertices. We have $m = \binom{n}{2}$ boolean variables, one for each potential edge. Then each boolean function $f : \{0,1\}^m \to \{0,1\}$ describes some graph property. A prominent NP-complete graph property is a monotone boolean function CLIQUE(n, k) which accepts a given graph on n vertices iff it contains a k-clique, that is, a subgraph on k vertices whose all vertices are pairwise adjacent. Instead of proving a lower bound on this function we will do this for a much larger class of "clique-like" functions.

An *a-coclique* is formed by assigning each vertex a color from the set $\{1, 2, \ldots, a\}$, and putting edges between those pairs of vertices with different colors. Note that no such graph can have an $(a+1)$-clique.

Let $2 \leq a < b \leq m$ be integers. An (a, b)-*clique function* is a monotone boolean function f such that, for every graph G on m vertices,

$$f(G) = \begin{cases} 0 & \text{if } G \text{ is an } a\text{-coclique;} \\ 1 & \text{if } G \text{ is a } b\text{-clique;} \\ \text{any value} & \text{otherwise.} \end{cases}$$

Hence, CLIQUE(n, k) is an (a, b)-clique function with $a = k-1$ and $b = k$.

[1] We take $(2r)^{4r}$ instead of just r^{4s} in order to cover also the real-valued case.

Theorem 9.26. (Jukna 1999) *Let* $32 \le a < b \le n/32$, *and let* f *be an* (a,b)-*clique function. Then the minimal number of gates in a monotone real circuit computing* f *is exponential in* $\min\{a, n/b\}^{1/4}$.

Proof. Let f be an (a,b)-clique function. We are going to apply the refined version of the lower-bounds criterion (Theorem 9.21). To do this, we must first choose appropriate length measure μ for positive inputs an a length measure η for negative inputs.

What to take as positive inputs and how to measure their length is clear. All b-cliques are positive inputs for f. A natural measure for a clique S is to take

$$\mu(S) := \text{ the number of vertices touched by the edges in } S\,.$$

It is clear that $\mu(S)$ is a legal length measure:

$$\mu(S \cup \{e\}) \le \mu(S) + \mu(\{e\}) = \mu(S) + 2 \quad \text{and} \quad |S| \le \binom{\mu(S)}{2} < \mu(S)^2\,.$$

Our choice of negative inputs is also clear: we take all complements of a-cocliques. Each such complement G_h is defined by a coloring h of vertices in a colors: two vertices u and v are adjacent in G_h iff $h(u) = h(v)$. But what should we take as a length measure $\eta(S)$ of such graphs in this case?

Having a graph S of a given η-measure $\eta(S) = s$, we want that as few as possible a-colorings h can color the edges of S monochromatically, that is, color both endpoints of each edge $e \in S$ by the same color. If S is a tree with s vertices, then we could take the same measure $\eta(S) = \mu(S) =$ the number of vertices touched by the edges in S. Now, $G_h \supseteq S$ implies that h must assign the same color to all $s = \eta(S)$ vertices of S, and we can have at most $a \cdot a^{n-s} = a^{n-s+1}$ such colorings. Thus, if S is a connected graph then we could take $\eta(S)$ be the maximum number of edges in its spanning tree. For general (not necessarily connected) graphs we can do the same, and consider the measure:

$$\eta(S) = \text{maximum number of edges in a forest } F \subseteq S.$$

Since every tree with m edges has $m+1$ vertices, $\eta(S)$ is just the number of vertices minus the number of connected components in S. But is η a legal length measure? The first condition $\eta(S) \le \eta(S \cup \{e\}) \le \eta(S) + c$ clearly holds with $c = 1$. But does the second condition $|S| \le \eta(S)^2$ hold? To show that it does, let m be the number of vertices touched by edges in S, and suppose that S consists of k connected components, the i-th of which has m_i vertices. We may assume that $m_i \ge 2$ for all i. Then $(m_i - 1)^2 \ge \binom{m_i}{2}$ holds for all i, and we obtain that

$$\eta(S)^2 = \Big[\sum_{i=1}^{k}(m_i - 1)\Big]^2 \ge \sum_{i=1}^{k}(m_i - 1)^2 \ge \sum_{i=1}^{k}\binom{m_i}{2} \ge |S|\,.$$

Thus, both measures $\mu(S)$ and $\eta(S)$ are legal length measures. By Theorem 9.25 it remains to choose parameters r, s and to show that our function f can have an (r, s)-approximator of size t only if t is large enough. For this purpose, we set (with foresight):

$$r := \lfloor (a/32)^{1/4} \rfloor \quad \text{and} \quad s := \lfloor (n/32b)^{1/4} \rfloor .$$

According to Definition 9.24 we have only two possibilities, depending on what of the two of its items holds.

Case 1: (Positive inputs) There exist a set I of $|I| \leq s^2$ edges and a family $Q_1, \ldots, Q_L$ of $L \leq t(2s)^{4r}$ r-cliques such that every b-clique must either intersect the set I or contain at least one of the cliques Q_i.

At least $\binom{n}{b} - s^2\binom{n-2}{b-2} \geq \frac{1}{2}\binom{n}{b}$ of b-cliques must avoid a fixed set I of $|I| \leq s^2$ edges. Each of these b-cliques must contain at least one of r-cliques Q_i. Since only $\binom{n-r}{b-r}$ of b-cliques can contain one clique Q_i, and we only have $L \leq t(2s)^{4r}$ of the Q_i, in this case we have the lower bound

$$t \geq \frac{\frac{1}{2}\binom{n}{b}}{(2s)^{4r}\binom{n-r}{b-r}} = \left(\frac{n}{16s^4 b}\right)^{\Omega(r)} = 2^{\Omega(a^{1/4})} .$$

Case 2: (Negative inputs) Recall that negative inputs are graphs G_h corresponding to colorings h of vertices in a colors; two vertices u and v are adjacent in G_h iff $h(u) = h(v)$. Recall also that $\eta(S)$ is the maximum number $|F|$ of edges in a spanning forest $F \subseteq S$. Thus, in the second case of Definition 9.24 there must be a family $\mathcal{F}$ of $|\mathcal{F}| \leq t(2r)^{4s}$ forests with $|F| \geq s$ edges in each $F \in \mathcal{F}$ such that every graph G_h contains at least one of these forests. That is, for every coloring h, there must be at least one forest $F \in \mathcal{F}$ such that $h(u) = h(v)$ for all edges of F.

Fix one forest $F \in \mathcal{F}$, and let $T_1, \ldots, T_d$ be all its connected components (trees). All vertices in each of these trees must receive the same color. Since each tree T_i has $|T_i| + 1$ vertices, the total number of vertices in the forest F is $m = \sum_{i=1}^{d}(|T_i| + 1) = |F| + d \geq s + d$. There are a^d ways for the coloring h to color the trees T_i, and at most $a^{n-m} \leq a^{n-(s+d)}$ ways to color the remaining $n - m$ vertices. Thus, the number of graphs G_h containing one fixed forest $F \in \mathcal{F}$ does not exceed $a^d a^{n-(s+d)} = a^{n-s}$. Since we only have $|\mathcal{F}| \leq t(2r)^{4s}$ forests in $\mathcal{F}$, in this case we have the lower bound

$$t \geq \frac{a^n}{(2r)^{4s} a^{n-s}} = \left(\frac{a}{16r^4}\right)^s = 2^{(n/b)^{1/4}} . \qquad \square$$

As mentioned above, the class of clique-like functions includes some NP-complete problems, like CLIQUE(n, k). But the class of (a, b)-clique functions is much larger—so large that it also includes some graph properties computable by non-monotone circuits of polynomial size!

A *graph function* is a function φ assigning each graph G a real number $\varphi(G)$. Such a function φ is *clique-like* if

$$\omega(G) \leq \varphi(G) \leq \chi(G) ,$$

where $\omega(G)$ is the clique number, that is, the maximum number of vertices in a complete subgraph of G, and $\chi(G)$ is the chromatic number, that is, the smallest number of colors which is enough to color the vertices of G so that no adjacent vertices receive the same color.

Although we always have that $\omega(G) \leq \chi(G)$, the gap between these two quantities can be quite large: results of Erdős (1967) imply that the maximum of $\chi(G)/\omega(G)$ over all n-vertex graphs G has the order $\Theta(n/\log^2 n)$. So, at least potentially, the class of clique-like functions is large enough. And indeed, Tardos (1987) observed that this class includes not only NP-complete problems (like the clique function) but also some problems from P.

Lemma 9.27. (Tardos 1987) *There exists an explicit monotone clique-like graph function φ which is computable in polynomial time.*

Proof. In his seminal paper on Shannon-capacity of graphs Lovász (1979a) introduced the capacity $\vartheta(G)$. The function $\varphi'(G) := \vartheta(\overline{G})$, where $\overline{G}$ denotes the complement of G, is a monotone clique-like function. Grötschel et al. (1981) gave a polynomial time approximation algorithm for ϑ. That is, given a graph G and a rational number $\epsilon > 0$ the algorithm computes, in polynomial time, a function $g(G,\epsilon)$ such that

$$\vartheta(G) \leq g(G,\epsilon) \leq \vartheta(G) + \epsilon\,.$$

Now, for any $0 < \epsilon < 1/2$ the function $\lfloor g(\overline{G},\epsilon)\rfloor$ is a polynomial time computable clique-like function. This function might not be monotone. Let us therefore consider the monotone function

$$\varphi(G) = \lfloor g(\overline{G}, n^{-2}) + e(G)\cdot n^{-2}\rfloor\,,$$

where n is the number of vertices and $e(G)$ the number of edges in G. This is the desired monotone clique-like function computable in polynomial time. □

Fix k to be the square root of the number n of vertices, and let f_ϕ denote the monotone boolean function of $\binom{n}{2}$ boolean variables encoding the edges of a graph on n vertices, whose values are defined by

$$f_\phi(G) = 1 \quad \text{iff} \quad \varphi(G) \geq k\,.$$

Observe that $f_\phi(G) = 1$ if $\omega(G) \geq k$, and $f_\phi(G) = 0$ if $\chi(G) \leq k-1$. Thus, f_ϕ is a $(k-1,k)$-clique function. Theorem 9.26 and Lemma 9.27 immediately yield the following tradeoff between monotone real and non-monotone boolean circuits.

Theorem 9.28. *~~For every clique-like graph function φ,~~ the boolean function f_ϕ can be computed by a non-monotone boolean circuit of polynomial size, but any monotone real circuit requires $2^{\Omega(n^{1/8})}$ gates.*

Thus, there are explicit monotone boolean functions, whose boolean non-monotone circuits are exponentially smaller than their monotone real circuits. We will use this

theorem later in Sect. 19.4 to prove exponential lower bounds for widely used proof system—cutting plane proofs.

But what about the other direction: ~~can every non-monotone boolean circuit computing a monotone boolean function be transformed into a monotone real circuit~~ without an exponential blow-up in size? Using counting arguments one can give a *negative* answer (see Exercises 9.4–9.6).

9.9 What About Circuits with NOT Gates?

As we mentioned at the very beginning, no non-linear lower bounds are known for circuits using NOT gates. So, what is missing in the arguments we described in this and the previous chapters?

A possible answer is that the arguments are just too general! In order to show that no circuit with t gates can compute a given boolean function f, we have to show that no such circuit C can separate the set $f^{-1}(0)$ from $f^{-1}(1)$, that is, reject *all* vectors in $f^{-1}(0)$ and accept *all* vectors in $f^{-1}(1)$. Current arguments for monotone circuits (and formulas) do much more: there are relatively small subsets $A \subseteq f^{-1}(0)$ and $B \subseteq f^{-1}(1)$ (sets of particular negative and positive inputs) such that every monotone circuit separating A from B must be large.

To be more specific, let A be the set of all complete $(k-1)$-partite graphs on n vertices, and B be the set of all k-cliques. Hence, for *any* k-clique function f, members of A are negative inputs and members of B are positive inputs for f. We have shown that any monotone circuit separating A from B must have exponential size.

On the other hand, A *can* be separated from B by a small circuit if we allow just one NOT gate be used at the top of the circuit! Indeed, each graph in A has at least $K = \Omega(n)$ edges, whereas each graph in B (a k-clique) has only $\binom{k}{2}$ edges, which is smaller than K for $k = o(\sqrt{n})$. Hence, if $g = \neg Th_K$ is the negation of the threshold-K function, then $g(a) = 0$ for all $a \in A$, and $g(b) = 1$ for all $b \in B$. Since threshold functions have small monotone circuits (at most quadratic in the number of input variables), the resulting circuit is also small, separates A from B, and has only one NOT gate.

That is, it is not hard to separate the pair A, B by a monotone circuit—it is only hard to do this separation in the "right" direction: reject all $a \in A$, and accept all $b \in B$. This motivates the following definition.

Let f be a monotone boolean function. Say that a pair A, B with $A \subseteq f^{-1}(0)$ and $B \subseteq f^{-1}(1)$ is r-*hard* if every monotone circuit separating a 2^{-r} fraction of A from a 2^{-r} fraction of B (either in a "right" or in a "wrong" direction) must have super-polynomial size.

Exercise 9.7 shows that any r-hard pair A, B requires a super-polynomial number of gates in any circuit that separates A from B and uses up to r NOT gates. In the next chapter we will show that $r = \lceil \log(n+1) \rceil$ is a critical number of allowed NOT gates: having an r-hard pair for such an r would imply a super-polynomial lower bound for general non-monotone circuits! The best result known today is that

the clique function produces an r-hard pair for r about $\log\log n$; this was shown by Amano and Maruoka (2005).

■ **Research Problem 9.29.** Exhibit an explicit pair A, B of disjoint subsets of $\{0,1\}^n$ which is r-hard for $r \gg \log\log n$.

9.10 Razborov's Method of Approximations

To describe the Method of Approximations in its full generality, it will again be convenient to look at boolean functions $f : \{0,1\}^X \to \{0,1\}$ as computing set-theoretic predicates $f : 2^X \to \{0,1\}$. In this way we get a 1-to-1 correspondence between boolean functions f and families $A(f) = \{S \subseteq X \mid f(S) = 1\}$ of subsets of X with the properties $A(f \vee g) = A(f) \cup A(g)$ and $A(f \wedge g) = A(f) \cap A(g)$. If f is monotone, then $A(f)$ is monotone with respect to set inclusion: if $E \in A(f)$ and $E \subseteq F$ then $F \in A(f)$.

Every family $\mathcal{F} \subseteq 2^X$ can be extended to a monotone family $\ulcorner \mathcal{F} \urcorner$ defined by

$$\ulcorner \mathcal{F} \urcorner := \bigcup_{F \in \mathcal{F}} \ulcorner F \urcorner, \quad \text{where} \quad \ulcorner F \urcorner := \{E \subseteq X \mid F \subseteq E\}.$$

In particular, if $F = \emptyset$ is the empty set, then $\ulcorner F \urcorner = 2^X$, whereas $\ulcorner \mathcal{F} \urcorner = \emptyset$ (empty family), if $\mathcal{F} = \emptyset$. The reason to consider monotone families is that we only consider monotone boolean functions f, and for them we have that $\ulcorner A(f) \urcorner = A(f)$.

Thus, each monotone circuit for a monotone boolean function f starts with the basic monotone families $A(x_1), \ldots, A(x_n), A(1) = 2^X, A(0) = \emptyset$ corresponding to input variables and the two constant functions, applies set-theoretic union ($\cup$) and intersection ($\cap$) operations to them, and finally produces the family $A(f)$. The idea is now to approximate the operations $\cup$ and $\cap$ by some other set-theoretic operations $\sqcup$ and $\sqcap$. This leads to the following definition.

A collection $\mathfrak{M} \subseteq 2^X$ of monotone families with two operations $\sqcup$ (join) and $\sqcap$ (meet) is a *legitimate lattice* if it satisfies the following two conditions:

1. Families $A(x_1), \ldots, A(x_n), A(1), A(0)$ belong to $\mathfrak{M}$.
2. $\mathfrak{M}$ is a lattice with respect to set inclusion, that is, $M, N \subseteq M \sqcup N$ and $M \sqcap N \subseteq M, N$ for all $M, N \in \mathfrak{M}$.

Note that the second condition implies that

$$M \cup N \subseteq M \sqcup N \quad \text{and} \quad M \sqcap N \subseteq M \cap N\,.$$

Thus, if we replace the gates $\cup$ and $\cap$ in our circuit by the lattice operations $\sqcup$ and $\sqcap$, then some element $M \in \mathfrak{M}$ instead of the target family $A(f)$ could be computed. To capture the errors arising at each gate, define:

$$\delta_-(M,N) := (M \sqcup N) \setminus (M \cup N),$$
$$\delta_+(M,N) := (M \cap N) \setminus (M \sqcap N).$$

Define the distance $\rho(f,\mathfrak{M})$ of a boolean function f from a lattice $\mathfrak{M}$ as the smallest number t for which there exist elements M, M_i, N_i $(1 \le i \le t)$ of the lattice $\mathfrak{M}$ such that

$$M \setminus A(f) \subseteq \delta_-(M_1,N_1) \cup \cdots \cup \delta_-(M_t,N_t),$$
$$A(f) \setminus M \subseteq \delta_+(M_1,N_1) \cup \cdots \cup \delta_+(M_t,N_t).$$

The proof of the following theorem is by easy induction on the number of gates, and we leave it as an exercise.

Theorem 9.30. *For every legitimate lattice $\mathfrak{M}$, every monotone boolean circuit computing f requires at least $\rho(f,\mathfrak{M})$ gates.*

In order to apply this theorem for a given monotone boolean function f, we have to define an appropriate legitimate lattice $\mathfrak{M}$ and show that f has a large distance from this lattice.

If we take $\mathfrak{M}$ to be a trivial lattice consisting of *all* monotone families, then $\rho(f,\mathfrak{M}) = 0$ for any monotone boolean function. So, in order to have a nontrivial distance, one has to consider some nontrivial lattices. For this, we need to achieve the following two goals:

1. Every family $M \in \mathfrak{M}$ must differ from $A(f)$ in many members.
2. The "error-families" $\delta_-(M_i,N_i)$ and $\delta_+(M_i,N_i)$ must be relatively small.

Crucial here is the second goal. Razborov achieves this goal by ensuring that each family in $\mathfrak{M}$ has relatively few minimal (w.r.t. set-inclusion) members. This, in turn, is achieved by introducing a clever "closure" operation, and by applying this operation when the union of two families in $\mathfrak{M}$ has too many minimal members.

9.10.1 Construction of Legitimate Lattices

Let $r \ge 2$ a fixed integer. Say that sets $F_1,\ldots,F_r$ imply a set F_0, and write $F_1,\ldots,F_r \vdash F_0$, if $F_i \cap F_j \subseteq F_0$ for all $1 \le i < j \le r$. We write $\mathcal{F} \vdash F$ if there exist not necessarily distinct members $F_1,\ldots,F_r$ of $\mathcal{F}$ such that $F_1,\ldots,F_r \vdash F$.

A general construction of legitimate lattices is as follows.

1. Fix an appropriate "ambient" family $\mathcal{P} \subseteq 2^X$. In the case of the clique function a natural choice is the family of all cliques on $\le s$ vertices, whereas in the case of the perfect matching function such is the family of all matchings with $\le s$ edges; s is a parameter.
2. Say that a family $\mathcal{F} \subseteq \mathcal{P}$ is *r-closed* (or just *closed*) if $\mathcal{F} \vdash F$ and $F \in \mathcal{P}$ implies $F \in \mathcal{F}$.
3. Define $\mathfrak{M} = \{\ulcorner \mathcal{A} \urcorner \mid \mathcal{A} \subseteq \mathcal{P}$ and $\mathcal{A}$ is r-closed$\}$.

Since the intersection of closed families is also closed, there is the smallest closed family containing $\mathcal{A}$, which we will denote by $\mathcal{A}^\star$.

Lemma 9.31. *For every family $\mathcal{P} \subseteq 2^X$, $\mathfrak{M}$ is a legitimate lattice with lattice operations given by*

$$\ulcorner \mathcal{A} \urcorner \sqcap \ulcorner \mathcal{B} \urcorner = \ulcorner \mathcal{A} \cap \mathcal{B} \urcorner \quad \textit{and} \quad \ulcorner \mathcal{A} \urcorner \sqcup \ulcorner \mathcal{B} \urcorner = \ulcorner (\mathcal{A} \cup \mathcal{B})^\star \urcorner .$$

Proof. First note that the condition (a) in the definition of a legitimate lattice is fulfilled: we have $A(x_i) = \ulcorner \{x_i\} \urcorner$ $A(1) = \ulcorner \mathcal{P} \urcorner$ and $A(0) = \ulcorner \emptyset \urcorner$.

Let $\mathfrak{A}$ denote the set of all r-closed families $\mathcal{A} \subseteq \mathcal{P}$. As the partially ordered with respect to the set-inclusion set, the set $\mathfrak{A}$ is a lattice with $\inf(\mathcal{A}_1, \mathcal{A}_2) = \mathcal{A}_1 \cap \mathcal{A}_2$ (intersection of two closed families is closed) and $\sup(\mathcal{A}_1, \mathcal{A}_2) = (\mathcal{A}_1 \cup \mathcal{A}_2)^*$. The mapping $\ulcorner \urcorner : \mathfrak{A} \to \mathfrak{M}$ is a homomorphism of partially ordered under set-inclusion sets. So, to finish the proof of the lemma, it is enough to show that this mapping is in fact an isomorphism. That is, to show that $\ulcorner \mathcal{A}_1 \urcorner \subseteq \ulcorner \mathcal{A}_2 \urcorner$ implies $\mathcal{A}_1 \subseteq \mathcal{A}_2$.

To show this, let $\ulcorner \mathcal{A}_1 \urcorner \subseteq \ulcorner \mathcal{A}_2 \urcorner$ and $E_1 \in \mathcal{A}_1$. Then $E_1 \in \ulcorner \mathcal{A}_1 \urcorner$, and hence, $E_1 \in \ulcorner \mathcal{A}_2 \urcorner$. That is, there must exist a set $E_2 \in \mathcal{A}_2$ such that $E_2 \subseteq E_1$. But then $E_2, \ldots, E_2 \vdash E_1$, implying that $E_1 \in \mathcal{A}_2$, since $\mathcal{A}_2$ is r-closed. We have therefore shown that $\ulcorner \mathcal{A}_1 \urcorner \subseteq \ulcorner \mathcal{A}_2 \urcorner$ implies $\mathcal{A}_1 \subseteq \mathcal{A}_2$, as desired. □

The main property of closed families is that they cannot have too many *minimal* members with respect to set-inclusion.

A set family $\mathcal{F}$ is an *antichain* if for no distinct A, B in $\mathcal{F}$ do we have $A \subset B$. For a family $\mathcal{F}$, let $\min(\mathcal{F})$ denote the antichain consisting of all smallest members of $\mathcal{F}$ with respect to set-inclusion.

Lemma 9.32. *If $\mathcal{F}$ is r-closed and $|F| \leq s$ for all $F \in \mathcal{F}$, then $|\min(\mathcal{F})| \leq s! r^s$.*

Proof. Assume that $|\min(\mathcal{F})| > s! r^s$. Then the Sunflower Lemma (applied with $l = s$ and $p = r + 1$) gives us $r + 1$ sets $F_0, F_1, \ldots, F_r$ in $\min(\mathcal{F})$ forming a sunflower. Since $\mathcal{F}$ is an antichain, the core E of this sunflower is a proper subset of each of the F_i, and hence, also $E \subset F_0$. But $F_i \cap F_j = E$ for all $1 \leq i < j \leq r$ implies that $F_1, \ldots, F_r \vdash E$, and hence, E must be a member of $\mathcal{F}$ since $\mathcal{F}$ is r-closed. This contradicts our assumption that $F_0 \in \min(\mathcal{F})$. □

9.11 A Lower Bound for Perfect Matching

The *perfect matching function* is a monotone boolean function f_m of m^2 variables. Inputs for this function are subsets $E \subseteq K_{m,m}$ of edges of a fixed complete bipartite $m \times m$ graph $K_{m,m}$, and $f_m(E) = 1$ iff E contains a perfect matching, that is, a set of m vertex-disjoint edges. Taking a boolean variable $x_{i,j}$ for each edge of $K_{m,m}$, the function can be written as

$$f_m = \bigvee_{\sigma \in S_m} \bigwedge_{i=1}^{m} x_{i,\sigma(i)} ,$$

where S_m is the set of all $m!$ permutations of $1, 2, \ldots, m$. The function f_m is also known as a *logical permanent* of a boolean $m \times m$ matrix, the adjacency matrix of E. Hopcroft and Karp (1973) showed that this sequence of functions $(f_m \mid m = 1, 2, \ldots)$ can be computed by a deterministic Turing machine in time $\mathcal{O}(m^{5/2})$. Hence, f_m can be computed by a non-monotone circuit using only $\mathcal{O}(m^5)$ gates. But what about *monotone* circuits for this function?

Using his Method of Approximations, Razborov (1985b) was able to prove a super-polynomial lower bound $m^{\Omega(\log m)}$ also for this function. Fu (1998) showed that, after an appropriate modification, Razborov's proof works also for monotone real circuits.

The lattice $\mathfrak{M}_m$ with large distance $\rho(f_m, \mathfrak{M}_m)$ from f_m will depend on two parameters r and s which we will set later. Namely, let $\mathfrak{M}_m$ be the lattice constructed as above when starting with the ambient family $\mathcal{P} = \mathrm{Per}_s$, where

$$\mathrm{Per}_s = \{E \subseteq K_{m,m} \mid E \text{ is a matching and } |E| \leq s\}$$

is the set of all matchings with up to s edges. That is, each element of $M \in \mathfrak{M}_m$ is produced by taking an r-closed collection $\mathcal{A} \subseteq \mathrm{Per}_s$ of matchings, each with $\leq s$ edges, and including in $\mathfrak{M}_m$ the monotone family $M = \ulcorner \mathcal{A} \urcorner$ of all graphs (not just matchings) containing at least one matching in $\mathcal{A}$. In particular, minimal (under inclusion) members of each M are matchings of size at most s, that is, $\min(M) \subseteq \mathrm{Per}_s$.

Our goal is to prove that, for appropriately chosen parameters r and s, we have $\rho(f_m, \mathfrak{M}_m) = m^{\Omega(\log m)}$.

It will be convenient to use probabilistic language. Let E_+ be a random graph taking its values in the set of all $m!$ perfect matchings with equal probability $1/m!$. It is clear that

$$\mathrm{Prob}[f_m(\mathrm{E}_+) = 1] = 1 .$$

Let h be a random two-coloring assigning each vertex of $K_{m,m}$ a value 0 or 1 independently with probability $1/2$. This coloring defines a random graph $\mathrm{E}_- = \{(u, v) \mid \mathrm{h}(u) = \mathrm{h}(v)\}$.

Lemma 9.33.

$$\mathrm{Prob}[f_m(\mathrm{E}_-) = 0] \geq 1 - \frac{2}{\sqrt{m}} .$$

Proof. Let U and V be the two parts of $K_{m,m}$; hence, $|U| = |V| = m$. The graph E_- has a perfect matching iff $\sum_{u \in U} \mathrm{h}(u) = \sum_{v \in V} \mathrm{h}(v)$. Hence,

$$\mathrm{Prob}[f_m(\mathrm{E}_-) = 1] = \mathrm{Prob}\Big[\sum_{u \in U} \mathrm{h}(u) = \sum_{v \in V} \mathrm{h}(v)\Big]$$

$$= \sum_{j=0}^{m} \text{Prob}\Big[\sum_{u\in U} \text{h}(u) = j\Big] \cdot \text{Prob}\Big[\sum_{v\in V} \text{h}(v) = j\Big]$$

$$\leq \max_{0\leq j\leq m} \text{Prob}\Big[\sum_{v\in V} \text{h}(v) = j\Big] \leq \binom{m}{m/2} \cdot 2^{-m} \leq \frac{2}{\sqrt{m}}\,. \qquad \square$$

In order to show that the distance $\rho(f_m, \mathfrak{M}_m)$ is large, it is enough to show that, for every two members M_1, M_2 of the lattice $\mathfrak{M}_m$, the probabilities $\text{Prob}[\text{E}_+ \in \delta_+(M_1, M_2)]$ and $\text{Prob}[\text{E}_- \in \delta_-(M_1, M_2)]$ are small.

9.11.1 Error-Probability on Accepted Inputs

The case of E_+ is relatively simple. Recall that E_+ is a random perfect matching.

Lemma 9.34. *For any $M_1, M_2 \in \mathfrak{M}_m$ we have that*

$$\text{Prob}[\text{E}_+ \in \delta_+(M_1, M_2)] \leq (s!r^s)^2 \cdot \frac{(m-s-1)}{m!}\,.$$

Proof. Let $M_1 = \ulcorner \mathcal{A}_1 \urcorner$ and $M_2 = \ulcorner \mathcal{A}_2 \urcorner$. Since for any family $\mathcal{F}$ and any two sets A, B we have that $\ulcorner \min(\mathcal{F}) \urcorner = \ulcorner \mathcal{F} \urcorner$ and $\ulcorner A \urcorner \cap \ulcorner B \urcorner = \ulcorner A \cup B \urcorner$, the error-set

$$\delta_+(M_1, M_2) = (M_1 \cap M_2) \setminus (M_1 \sqcap M_2) = (\ulcorner \mathcal{A}_1 \urcorner \cap \ulcorner \mathcal{A}_2 \urcorner) \setminus (\ulcorner \mathcal{A}_1 \cap \mathcal{A}_2 \urcorner)$$

is the union of sets $\ulcorner E_1 \cup E_2 \urcorner \setminus (\ulcorner \mathcal{A}_1 \cap \mathcal{A}_2 \urcorner)$ over all $E_1 \in \min(\mathcal{A}_1)$ and $E_2 \in \min(\mathcal{A}_2)$. Fix any two such sets E_1 and E_2, and let $E = E_1 \cup E_2$. Our goal is to upper-bound the probability $\text{Prob}[\text{E}_+ \in \ulcorner E \urcorner] = \text{Prob}[E \subseteq \text{E}_+]$. We have three possibilities.

Case 1: E is not a matching. In this case $\text{Prob}[E \subseteq \text{E}_+] = 0$.

Case 2: E is a matching and $|E| \leq s$, that is, $E \in \text{Per}_s$. Since $\mathcal{A}_1$ is closed, $E_1 \in \mathcal{A}_1$ and $E \in \text{Per}_s$ implies that $E = E_1 \cup E_2 \in \mathcal{A}_1$. Similarly, $E \in \mathcal{A}_2$. Hence $E \in \mathcal{A}_1 \cap \mathcal{A}_2$, implying that $\ulcorner E_1 \cup E_2 \urcorner \setminus (\ulcorner \mathcal{A}_1 \cap \mathcal{A}_2 \urcorner) = \emptyset$.

Case 3: E is a matching but $|E| \geq s+1$. In this case

$$\text{Prob}[E \subseteq \text{E}_+] = \frac{(m-|E|)!}{m!} \leq \frac{(m-s-1)!}{m!}\,.$$

Since, by Lemma 9.32, $|\min(\mathcal{A}_1)| \cdot |\min(\mathcal{A}_1)| \leq (s!r^s)^2$, we are done. $\square$

9.11.2 Error-Probability on Rejected Inputs

To upper bound the probability Prob[$E_- \in \delta_-(M_1, M_2)$] requires more work. The problem is that the events $e_1 \in E_-$ and $e_2 \in E_-$ for edges e_1, e_2 are not necessarily independent. Still, the following lemma shows that the events are independent if the edges come from a fixed forest. Recall that a *forest* is a graph without cycles.

Lemma 9.35. *Let $E = \{(u_1, v_1), \ldots, (u_p, v_p)\} \subseteq K_{m,m}$ be a forest. Then the events $(u_i, v_i) \in E_-$ are independent, and each happens with probability $1/2$.*

Proof. It is enough to show that, for any subset $K \subseteq \{1, \ldots, p\}$ of indices, the event

$$(u_i, v_i) \in E_- \text{ for all } i \in K, \text{ and } (u_j, v_j) \notin E_- \text{ for all } j \notin K$$

happens with probability 2^{-p}. By the definition of E_-, this event is equivalent to the event that the values $h(u_i), h(v_i)$ satisfy the following system of linear equations over GF(2):

$$h(u_i) + h(v_i) = \chi_K + 1 \quad i = 1, \ldots, p\,, \tag{9.7}$$

where $h(u_i), h(v_i)$ are treated as variables, and χ_K is the characteristic function of the set K. Since E is a forest, the left-hand side of this system is linearly independent (see Exercise 9.8). Thus, the system has exactly 2^{2m-p} solutions, as desired. □

Lemma 9.36. *Let $\mathcal{F} \subseteq \mathrm{Per}_s$ be a set of $|\mathcal{F}| = r$ pairwise disjoint matchings. Then there exits a subset $\mathcal{F}_0 \subseteq \mathcal{F}$ of $|\mathcal{F}_0| \geq \sqrt{r}/s$ matchings such that $\cup\mathcal{F}_0$ is a forest.*

Proof. Choose $\mathcal{F}_0 \subseteq \mathcal{F}$ such that $\cup\mathcal{F}_0$ is a forest and $|\mathcal{F}_0|$ is maximal. It is enough to show that $|\mathcal{F}_0| \geq \sqrt{r}/s$.

To show this, assume that $|\mathcal{F}_0| < \sqrt{r}/s$, and let $E_0 = \cup\mathcal{F}_0$; hence, $|E_0| < \sqrt{r}$. Let $U_0 \subseteq U$ and $V_0 \subseteq V$ be the sets of vertices incident with at least one edge of E_0. Then $|U_0| < \sqrt{r}$ and $|V_0| < \sqrt{r}$. Since $\mathcal{F}$ contains $|\mathcal{F}| = r > |U_0 \times V_0|$ matchings, at least one of these matchings E_1 must have no edge in $U_0 \times V_0$ (every edge can belong to at most one matching in $\mathcal{F}$, since these matchings are disjoint). Since E_1 is a matching and E_0 is a forest lying in $U_0 \times V_0$, the graph $E_0 \cup E_1$ is a forest as well. But $E_1 \cap E_0 = \emptyset$ implies that $E_1 \notin \mathcal{F}_0$, a contradiction with the maximality of $|\mathcal{F}_0|$. □

Now we are able to upper-bound Prob[$E_- \in \delta_-(M_1, M_2)$]. Note that the number of matchings in Per_s is

$$|\mathrm{Per}_s| \leq \sum_{i=0}^{s} \binom{m}{i}^2 \cdot i\,! \leq m^s \sum_{i=0}^{s} \binom{m}{i} \leq m^{2s}\,.$$

Lemma 9.37. *For any $M_1, M_2 \in \mathfrak{M}_m$ we have that*

$$\mathrm{Prob}[E_- \in \delta_-(M_1, M_2)] \leq (1 - 2^{-s})^{\sqrt{r}/s} \cdot m^{2s}\,.$$

Proof. Let $M_1 = \ulcorner \mathcal{A}_1 \urcorner$, $M_2 = \ulcorner \mathcal{A}_2 \urcorner$ and $\mathcal{A}_3 = \mathcal{A}_1 \cup \mathcal{A}_2$. Then $\delta_-(M_1, M_2) = \ulcorner \mathcal{A}_3^\star \urcorner \setminus \ulcorner \mathcal{A}_3 \urcorner$, where $\mathcal{A}_3^\star$ is the closure of $\mathcal{A}_3$. Hence, there is a sequence of families $\mathcal{A}_3, \mathcal{A}_4, \ldots, \mathcal{A}_p = \mathcal{A}_3^\star$ such that $\mathcal{A}_{i+1} = \mathcal{A}_i \cup \{E_i\}$ with $\mathcal{A}_i \vdash E_i$ and $E_i \notin \mathcal{A}_i$. Hence, $\delta_-(M_1, M_2)$ is the union of all sets $\ulcorner E_i \urcorner \setminus \ulcorner \mathcal{A}_i \urcorner$, $i = 3, \ldots, p-1$. Since $p \le |\mathrm{Per}_s| \le m^{2s}$, it remains to show that $\mathcal{A} \subseteq \mathrm{Per}_s$ and $\mathcal{A} \vdash E_0$ implies that

$$\mathrm{Prob}[\mathrm{E}_- \in \ulcorner E_0 \urcorner \setminus \ulcorner \mathcal{A} \urcorner] \le (1 - 2^{-s})^{\sqrt{r}/s} . \tag{9.8}$$

To prove this, let $E_1, \ldots, E_r$ be matchings in $\mathcal{A}$ such that $E_1, \ldots, E_r \vdash E_0$. Hence, the sets $E_i^* := E_i \setminus E_0$ must be disjoint. If at least one of these sets is empty, then $\ulcorner E_0 \urcorner \subseteq \ulcorner \mathcal{A} \urcorner$, and the inequality (9.8) trivially holds. Otherwise, we can use Lemma 9.36 to choose a subset $\mathcal{F}_0 \subseteq \{E_1^*, \ldots, E_r^*\}$ such that $\cup \mathcal{F}_0$ is a forest and $|\mathcal{F}_0| \ge \sqrt{r}/s$. Then

$$\begin{aligned}
\mathrm{Prob}[\mathrm{E}_- \in \ulcorner E_0 \urcorner \setminus \ulcorner \mathcal{A} \urcorner] &\le \mathrm{Prob}[E_0 \subseteq \mathrm{E}_- \text{ and } E_i \not\subseteq \mathrm{E}_- \text{ for all } i = 1, \ldots, r\,] \\
&\le \mathrm{Prob}[E_i^* \not\subseteq \mathrm{E}_- \text{ for all } i = 1, \ldots, r\,] \\
&\le \mathrm{Prob}[E^* \not\subseteq \mathrm{E}_- \text{ for all } E^* \in \mathcal{F}_0\,] .
\end{aligned}$$

By Lemma 9.35, all events $E^* \not\subseteq \mathrm{E}_-$ for $E^* \in \mathcal{F}_0$ are independent, and

$$\mathrm{Prob}[E^* \subseteq \mathrm{E}_-] = 2^{-|E^*|} \ge 2^{-s} .$$

Therefore,

$$\mathrm{Prob}[E^* \not\subseteq \mathrm{E}_- \text{ for all } E^* \in \mathcal{F}_0\,] = \prod_{E^* \in \mathcal{F}_0} \mathrm{Prob}[E^* \not\subseteq \mathrm{E}_-] \le (1 - 2^{-s})^{\sqrt{r}/s} .$$

This finishes the proof of (9.8), and thus of the lemma. □

Theorem 9.38. (Razborov 1985b) *Every monotone circuit computing the perfect matching function f_m must have $m^{\Omega(\log m)}$ gates.*

Proof. By Theorem 9.30, it is enough to show that $\rho(f_m, \mathfrak{M}_m) = m^{\Omega(\log m)}$. For the proof we assume that m is sufficiently large, and set the parameters r and s to

$$s := \lfloor (\log m)/8 \rfloor \quad \text{and} \quad r := \lfloor m^{1/4} (\log m)^8 \rfloor .$$

Let M, M_i, N_i $(1 \le i \le t)$ be elements of the lattice $\mathfrak{M}_m$ such that

$$M \setminus A(f_m) \subseteq \bigcup_{i=1}^{t} \delta_-(M_i, N_i), \tag{9.9}$$

$$A(f_m) \setminus M \subseteq \bigcup_{i=1}^{t} \delta_+(M_i, N_i) . \tag{9.10}$$

We consider two cases; $M = \emptyset$ and $M \neq \emptyset$.

Case 1: $M = \emptyset$. In this case, (9.10) implies that the entire set $A(f_m)$ must lie in the union of error-sets $\delta_+(M_i, N_i)$, $i = 1, \ldots, t$. Since E_+ lies in $A(f_m)$ with probability 1, the sum of probabilities $\mathrm{Prob}[\mathrm{E}_+ \in \delta_+(M_i, N_i)]$ must be at least 1 as well. Together with Lemma 9.34, this implies that (for sufficiently large m)

$$\begin{aligned} t &\geq \frac{m!}{(m-s-1)!(s!r^s)^2} \geq (m/2)^s \cdot (sr)^{-2s} \\ &= \left(\frac{m}{2r^2s^2}\right)^s \\ &\geq \left(\frac{32m}{m^{1/2}(\log m)^{18}}\right)^{(\log m)/8} \\ &= m^{\Omega(\log m)} . \end{aligned}$$

Case 2: $M \neq \emptyset$. By the construction of $\mathfrak{M}_m$, there exists a matching $E \in \mathrm{Per}_s$ for which $\ulcorner E \urcorner \subseteq M$. Together with (9.9), this implies that

$$\ulcorner E \urcorner \subseteq A(f_m) \cup \delta_-(M_1, N_1) \cup \cdots \cup \delta_-(M_t, N_t) .$$

We have

$$\begin{aligned} \mathrm{Prob}[\mathrm{E}_- \in \ulcorner E \urcorner] &= 2^{-|E|} \geq 2^{-s} && \text{by Lemma 9.35} \\ \mathrm{Prob}[\mathrm{E}_- \in A(f_m)] &\leq 2m^{-1/2} && \text{by Lemma 9.33} \\ \mathrm{Prob}[\mathrm{E}_- \in \delta_-(M_i, N_i)] &\leq (1-2^{-s})^{\sqrt{r}/s} \cdot m^{2s} && \text{by Lemma 9.37} \end{aligned}$$

This implies that

$$\begin{aligned} t &\geq (2^{-s} - 2m^{-1/2})(1-2^{-s})^{-\sqrt{r}/s} m^{-2s} \\ &\geq \tfrac{1}{8} m^{-1/8} \cdot \exp\left(\frac{2^{-s}\sqrt{r}}{s}\right) \cdot m^{-2s} \\ &\geq \tfrac{1}{8} m^{-\frac{1}{8}-\frac{1}{4}\log m} \exp\left(\frac{8m^{-1/8} \cdot m^{1/8} \cdot (\log m)^4}{\log m}\right) \\ &= m^{\Omega(\log^3 m)} . \end{aligned}$$

□

■ **Research Problem 9.39.** Can the lower bound $m^{\Omega(\log m)}$ for perfect matching be improved to $2^{\Omega(m^\epsilon)}$ for a constant $\epsilon > 0$?

Exercises

9.1. A *partial* b–(n,k,λ) *design* is a family $\mathcal{F}$ of k-element subsets of $\{1,\dots,n\}$ such that any b-element set is contained in at most λ of its members. We can associate with each such design $\mathcal{F}$ a monotone boolean function $f_{\mathcal{F}}$ such that $f_{\mathcal{F}}(S)=1$ if and only if $S \supseteq F$ for at least one $F \in \mathcal{F}$. Assume that $\ln|\mathcal{F}| < k-1$ and that each element belongs to at most N members of $\mathcal{F}$. Use Theorem 9.17 to show that for every integer $a \geq 2$, every monotone circuit computing $f_{\mathcal{F}}$ has size at least

$$L := \min\left\{\frac{1}{2}\left(\frac{k}{2b\ln|\mathcal{F}|}\right)^a,\ \frac{|\mathcal{F}|-a\cdot N}{\lambda\cdot a^b}\right\}.$$

Hint: Take $r=a$, $s=b$ and show that under this choice of parameters, the function $f_{\mathcal{F}}$ can be t-simple only if $t \geq L$. When doing this, note that the members of $\mathcal{F}$ are positive inputs for $f_{\mathcal{F}}$. To handle the case of negative inputs, take a random subset T in which each element appears independently with probability $p=(1+\ln|\mathcal{F}|)/k$, and show that T is not a negative input for $f_{\mathcal{F}}$ with probability at most $|\mathcal{F}|(1-p)^k \leq \mathrm{e}^{-1}$.

9.2. Derive Theorem 9.20 from the previous exercise.

Hint: Observe that the family of all q^d graphs of polynomials of degree at most $d-1$ over GF(q) forms a partial b–(n,k,λ) design with parameters $n=q^2$, $k=q$ and $\lambda=q^{d-b}$.

9.3. Andreev (1987b) showed how, for any prime power $q \geq 2$ and $d \leq q$, to construct an explicit family $\mathcal{F}$ of subsets of $\{1,\dots,n\}$ which, for every $b \leq d+1$, forms a partial b–(n,k,λ) design with parameters $n=q^3$, $k=q^2$, $\lambda=q^{2d+1-b}$ and $|\mathcal{F}|=q^{2d+1}$. Use Exercise 9.1 to show that the corresponding boolean function $f_{\mathcal{D}}$ requires monotone circuits of size exponential in $\Omega(n^{1/3-o(1)})$.

9.4. A boolean function $f(x_1,\dots,x_n)$ is a *k-slice function* if $f(x)=0$ for all x with $|x|<k$, and $f(x)=1$ for all x with $|x|>k$, where $|x|=x_1+\cdots+x_n$. Show that some slice functions require DeMorgan circuits of size $2^{\Omega(n)}$.

Hint: Take $k=n/2$ and argue as in the proof of Theorem 1.14.

9.5. (Rosenbloom 1997) Given a vector $x=(x_1,\dots,x_n)$ in $\{0,1\}^n$, associate with it the following two integers $h_+(x) := |x|2^n + b(x)$ and $h_-(x) := |x|2^n - b(x)$, where $|x|=x_1+\cdots+x_n$ and $b(x)=\sum_{i=1}^n x_i 2^{i-1}$. Prove that for any two vectors $x \neq y$,

1. if $|x|<|y|$, then $h_+(x)<h_+(y)$ and $h_-(x)<h_-(y)$;
2. if $|x|=|y|$, then $h_+(x) \leq h_+(y)$ if and only if $h_-(x) \geq h_-(y)$.

9.6. Let $f(x_1,\dots,x_n)$ be a k-slice function, $0 \leq k \leq n$. Use the previous exercise to show that f can be computed by a circuit with $\mathcal{O}(n)$ monotone real-valued functions as gates.

Hint: As the last gate take a monotone function $\varphi : \mathbb{R}^2 \to \{0,1\}$ such that

$$\varphi(h_+(x),h_-(x)) = f(x)$$

for all inputs x of weight $|x|=k$.

9.7. Let f be a boolean function and suppose that it can be computed by a circuit of size t with at most r negations. Show that for any $A \subseteq f^{-1}(0)$ and $B \subseteq f^{-1}(1)$, there is a monotone boolean function g such that g can be computed by a monotone circuit of size at most t and either g or its negation $\neg g$ rejects a 2^{-r} fraction of inputs from A and accepts a 2^{-r} fraction of inputs from B.

Hint: Argue by induction on r. If $r \geq 1$, then consider the first negation gate and the function g which is computed at the gate immediately before this negation gate. Let $\epsilon \in \{0,1\}$ be such that $g(a) = \epsilon$ for at least one half of the inputs $a \in A$. If also one half of the inputs $b \in B$ have $g(b) = \epsilon \oplus 1$, then either g or $\neg g$ has the ~~property stated in the lemma~~. If this is not the case, try to apply the induction hypothesis.

9.8. Let G be a graph with n vertices and m edges, and let M be its $m \times n$ edge-vertex adjacency 0-1 matrix. That is, there is a 1 in the i-th row and j-th column iff the j-th vertex is an endpoint of the i-th edge. Show that the rows of M are linearly independent over GF(2) if and only if G is a forest.

Hint: In any non-empty forest there are at least two vertices of degree 1. If some subset of rows sums up to zero, then the subgraph formed by the corresponding edges must have minimum degree at least 2.

9.9. A set $A \subseteq \{0,1\}^n$ of vectors is Downward Closed if $x \in A$ and $y \leq x$ implies $y \in A$. Similarly, a set is Upward Closed if $x \in A$ and $x \leq y$ implies $y \in A$. Note that, if a boolean function $f : \{0,1\}^n \to \{0,1\}$ is monotone, then $f^{-1}(0)$ is Downward Closed and $f^{-1}(1)$ is Upward Closed. Prove the following result due to Kleitman (1966): if A, B are Downward Closed subsets of $\{0,1\}^n$, then

$$|A \cap B| \geq \frac{|A| \cdot |B|}{2^n}\,.$$

Hint: Apply induction on n, the case $n = 0$ being trivial. For $a \in \{0,1\}$, set $c_a = |A_a|$ and $d_a = |B_a|$, where $A_a = \{(x_1, \ldots, x_{n-1}) \mid (x_1, \ldots, x_{n-1}, a) \in A\}$. Apply induction to show that $|A \cap B| \geq (c_0 d_0 + c_1 d_1)/2^{n-1}$ and use the equality $c_0 d_0 + c_1 d_1 = (c_0 + c_1)(d_0 + d_1) + (c_0 - c_1)(d_0 - d_1)$ together with $A_1 \subseteq A_0$ and $B_1 \subseteq B_0$.

9.10. Show that Kleitman's theorem (Exercise 9.9) implies the following: Let A, B be upward closed and C downward closed subsets of $\{0,1\}^n$. Then

$$|A \cap B| \geq \frac{|A| \cdot |B|}{2^n} \qquad \text{and} \qquad |A \cap C| \leq \frac{|A| \cdot |C|}{2^n}\,.$$

Hint: For the first inequality, apply Kleitman's theorem to the complements of A and B. For the second inequality, take $B := \{0,1\}^n \setminus C$, and apply the first inequality to the pair A, B to get $|A| - |A \cap C| = |A \cap B| \geq 2^{-n}|A|(2^n - |C|)$.

9.11. Let $f : 2^{[n]} \to \{0,1\}$ be a monotone boolean function, and let $\mathcal{F}$ be the family of all subsets $S \subseteq [n]$ that are both positive and negative inputs of f, that is $f(S) = 1$ and $f(\overline{S}) = 0$. Show that $|\mathcal{F}| \leq |f^{-1}(0)| \cdot |f^{-1}(1)|/2^n$.

9.12. (Flower Lemma, Håstad et al. 1995) A *blocking set* of a family $\mathcal{F}$ is a set which intersects all the members of $\mathcal{F}$; the minimum number of elements in a

blocking set is the *blocking number* of $\mathcal{F}$ and is denoted by $\tau(\mathcal{F})$; if $\emptyset \in \mathcal{F}$ then we set $\tau(\mathcal{F}) = 0$. A *restriction* of a family $\mathcal{F}$ onto a set Y is the family $\mathcal{F}_Y := \{S \setminus Y \mid S \in \mathcal{F},\ S \supseteq Y\}$. A *flower* with k petals and a core Y is a family $\mathcal{F}$ such that $\tau(\mathcal{F}_Y) \geq k$. Note that every sunflower is a flower with the same number of petals, but not every flower is a sunflower (give an example). Prove the following "flower lemma":

Let $\mathcal{F}$ be a family of sets each of cardinality s, and $k \geq 1$ and integer. If $|\mathcal{F}| > (k-1)^s$ then $\mathcal{F}$ contains a flower with k petals.

Hint: Induction on s. If $\tau(\mathcal{F}) \geq k$ then the family $\mathcal{F}$ itself is a flower with at least $(k-1)^s + 1 \geq k$ petals (and an empty core). Otherwise, some set of size $k-1$ intersects all the members of $\mathcal{F}$, and hence, at least $|\mathcal{F}|/(k-1)$ of the members must contain some point x.

9.13. Let f be a monotone boolean function of n variables, and suppose that all its maxterms have length at most t. Show that then for every $s = 1, \ldots, n$ the function f has at most t^s minterms of length s.

Hint: Let $\mathcal{F}$ be the family of all minterms of f of length s. Every maxterm must intersect all the minterms in $\mathcal{F}$. Assume that $|\mathcal{F}| > t^s$ and apply the Flower Lemma to get a contradiction with the previous sentence.

9.14. Use Exercise 9.13 to give an alternate proof of the Monotone Switching Lemma.