Chapter 13

Lower Bounds

The very simple problem PARITY *is too hard for first-order logic, no matter what numeric predicates we add. When we add counting, but remove ordering,* PARITY *is expressible. However, a different sort of parity problem becomes inexpressible. A related lower bound suggests that complete problems for* P *are inherently sequential.*

13.1 Håstad's Switching Lemma

Recall boolean query PARITY, which is true of boolean strings that have an odd number of ones. Using pebble games, we have shown that PARITY is not first-order in the absence of the numeric predicate BIT (Proposition 6.14, Proposition 6.44). This theorem is much more subtle with the inclusion of BIT.

Theorem 13.1 *Query* PARITY *is not first-order expressible:* PARITY $\notin$ FO.

The known proofs of Theorem 13.1 all prove the stronger result that PARITY is not in the non-uniform class $AC^0/poly$ or, equivalently, PARITY is not first-order, no matter what numeric predicates are available (Proposition 11.19). The proof we present here is via the Håstad Switching Lemma, following the treatment in [Bea96].

Let f be a boolean function, with boolean variables $V_n = \{x_1, \ldots, x_n\}$. A *restriction* on V_n is a map $\rho : V_n \to \{0, 1, \star\}$. The idea is that some of the variables are set to "0" or "1" and the others — those assigned "$\star$" — remain variables.

Restriction ρ applied to function f results in function $f|_\rho$ in which value $\rho(x_i)$ is substituted for x_i in f, for each x_i such that $\rho(x_i) \neq \star$. Thus, $f|_\rho$ is a function of the variables that have been assigned "$\star$". Let $\mathcal{R}_n^r$ be the set of all restrictions on V_n that map exactly r variables to "$\star$".

We state and prove the switching lemma using decision trees. Given a formula F in disjunctive normal form (DNF)[1] define the *canonical decision tree* $T(F)$ *for* F as follows: Let $C_1 = \ell_1 \wedge \cdots \wedge \ell_i$ be the first term of F, so $F = C_1 \vee F'$. The top of $T(F)$ is a complete binary decision tree on the variables in C_1. Each leaf of the tree determines a restriction ρ that assigns the appropriate value to the variables in C_1 and assign "$\star$" to all the other variables. There is a unique leaf that makes C_1 true and this should remain a leaf and be labeled "1". To each other leaf, determining restriction ρ, we attach the canonical decision tree $T(F'|_\rho)$.

Let $h(T)$ be the height of tree T. We now show that for any formula F in DNF, if F has only small terms, then when randomly choosing a restriction ρ from $\mathcal{R}_n^r$, with high probability the height of the canonical decision tree of the resulting formula, $h(T(F|_\rho))$, is small.

It then follows that the negation of $F|_\rho$ can also be written in DNF — as the disjunction of the conjunction of each branch in the tree that leads to "0". Thus, with high probability, a random restriction switches a DNF formula that has only small terms to a conjunctive normal form (CNF) formula.

Lemma 13.2 (Håstad Switching Lemma) *Let F be a DNF formula on n variables, such that each of its terms has length at most k. Let $p \leq 1/7$, $r = pn$, and $s \geq 0$. Then,*

$$\frac{|\{\rho \in \mathcal{R}_n^r \,|\, h(T(F|_\rho)) \geq s\}|}{|\mathcal{R}_n^r|} \;<\; (7pk)^s \;.$$

Proof The proof of Lemma 13.2 is a somewhat intricate counting argument. Let Stars(k, s) be the set of all sequences $w = (S_1, S_2, \ldots, S_t)$ where each S_i is a nonempty subset of $\{1, 2, \ldots, k\}$ and the sum of the cardinalities of the S_i's equals s. We use the following upper bound on the size of Stars(k, s).

Lemma 13.3 *For $k, s > 0$,* $|\text{Stars}(k, s)| \;\leq\; (k/\ln 2)^s$.

[1] A DNF formula is an "or" of "and"s. This is the dual of CNF.

Proof We show by induction on s that $|\text{Stars}(k,s)| \leq \gamma^s$, where γ is such that $(1+1/\gamma)^k = 2$. Since $(1+1/\gamma) < e^{1/\gamma}$, we have $\gamma < k/\ln 2$ and thus the lemma will follow.

Suppose that the lemma holds for any $s' < s$. Let $\beta \in \text{Stars}(k,s)$. Then $\beta = (S_1, \beta')$, where $\beta' \in \text{Stars}(k, s-i)$ and $i = |S_1|$. Thus,

$$|\text{Stars}(k,s)| \quad = \quad \sum_{i=1}^{\min(k,s)} \binom{k}{i} |\text{Stars}(k, s-i)|$$

Thus, by the induction hypothesis,

$$\begin{aligned} |\text{Stars}(k,s)| &\leq \sum_{i=1}^{k} \binom{k}{i} \gamma^{s-i} \\ &= \gamma^s \sum_{i=1}^{k} \binom{k}{i} (1/\gamma)^i \\ &= \gamma^s[(1+1/\gamma)^k - 1] \quad = \quad \gamma^s \ . \end{aligned}$$

□

Let $R \subseteq \mathcal{R}_n^r$ be the set of restrictions ρ such that $h(T(F|_\rho)) \geq s$. We will define a 1:1 map,

$$\alpha : R \to \mathcal{R}_n^{r-s} \times \text{Stars}(k,x) \times 2^s \ . \tag{13.4}$$

Once we show that α is one to one, it will follow that

$$\frac{|R|}{|\mathcal{R}_n^r|} \ \leq \ \frac{|\mathcal{R}_n^{r-s}|}{|\mathcal{R}_n^r|} \cdot |\text{Stars}(k,s)| \cdot 2^s \ . \tag{13.5}$$

Observe that $|\mathcal{R}_n^r| = \binom{n}{r} 2^{n-r}$, so,

$$\frac{|\mathcal{R}_n^{r-s}|}{|\mathcal{R}_n^r|} \ = \ \frac{(r)(r-1)\cdots(r-s+1)}{(n-r+s)(n-r+s-1)\cdots(n-r+1)} \cdot 2^s \ \leq \ \left(\frac{2r}{n-r}\right)^s \ .$$

Substituting this into Equation (13.5) and using Lemma 13.3, we have,

$$\begin{aligned} \frac{|R|}{|\mathcal{R}_n^r|} &\leq \left(\frac{2r}{n-r}\right)^s \cdot (k/\ln 2)^s \cdot 2^s \\ &= \left(\frac{4rk}{(n-r)\ln 2}\right)^s \\ &= \left(\frac{4pk}{(1-p)\ln 2}\right)^s \end{aligned}$$

when $r = pn$. This is less than $(7pk)^s$ when $p < 1/7$.

It thus suffices to construct 1:1 map α (Equation (13.4)). Let $F = C_1 \vee C_2 \vee \cdots$. Let $\rho \in R$, and let C_{i_1} be the first term of F that is not set to "0" in $F|_\rho$.

Let b be the first s steps of the lexicographically first branch in $T(F|_\rho)$ that has length at least s. Let V_1 be the set of variables in $C_{i_1}|_\rho$. Let a_1 be the assignment to V_1 that makes $C_{i_1}|_\rho$ true. Let b_1 be the initial segment of b that assigns values to V_1. If b ends before all the values of V_1 are defined, then let $b_1 = b$, and shorten a_1 so that it assigns values only to the variables that b_1 does. See Figure 13.6.

Define the set $S_1 \subseteq \{1, 2, \ldots, k\}$ to include those j such that the j^{th} variable in C_{i_1} is set by a_1. S_1 is nonempty. Note that from C_{i_1} and S_1 we can reconstruct a_1.

If $b \neq b_1$, then $(b - b_1)$ is a path in $T(F|_{\rho b_1})$. Let C_{i_2} be the first term of F not set to "0" by ρb_1. As above, we generate b_2, a_2, and S_2. Repeat this until the whole branch b is used up. We have $b = b_1 b_2 \cdots b_t$, and let $a = a_1 a_2 \cdots a_t$. Define the map $\delta : \{1, \ldots, s\} \to \{0, 1\}$ such that $\delta(j) = 1$ if a and b assign the same value at their step j, and $\delta(j) = 0$ if a and b assign different values to variable j. We finally define the map α as,

$$\alpha(\rho) \quad = \quad \langle \rho a, (S_1, S_2, \ldots, S_t), \delta \rangle \ .$$

From $\alpha(\rho)$ we can reconstruct ρ as follows: C_{i_1} is the first clause that evaluates to "1" using ρa. From C_{i_1} and S_1 we reconstruct a_1. Then, using δ, we can compute the restriction $\rho' = \rho b_1 a_2 \cdots a_t$. Next, C_{i_2} is the first clause evaluating to "1" using ρ'. From this and S_2, we can compute a_2, and so on. Thus α is 1:1. This completes the proof of Håstad's Switching Lemma. □

A striking consequence of the switching lemma is that AC^0 circuits have restrictions on which they are constant even though many variables are assigned to "$\star$":

Theorem 13.7 *Let C be an unbounded fan-in circuit with n inputs, having size s and depth d. Let $r \leq n/(14^d(\log s)^{d-1}) - (\log(s) - 1)$. Then there is a restriction $\rho \in \mathcal{R}_n^r$ for which $C|_\rho$ is constant.*

Proof We show inductively from the leaves up, that there is a restriction that turns all the gates into DNF or CNF formulas all of whose terms have length at most $\log s$.

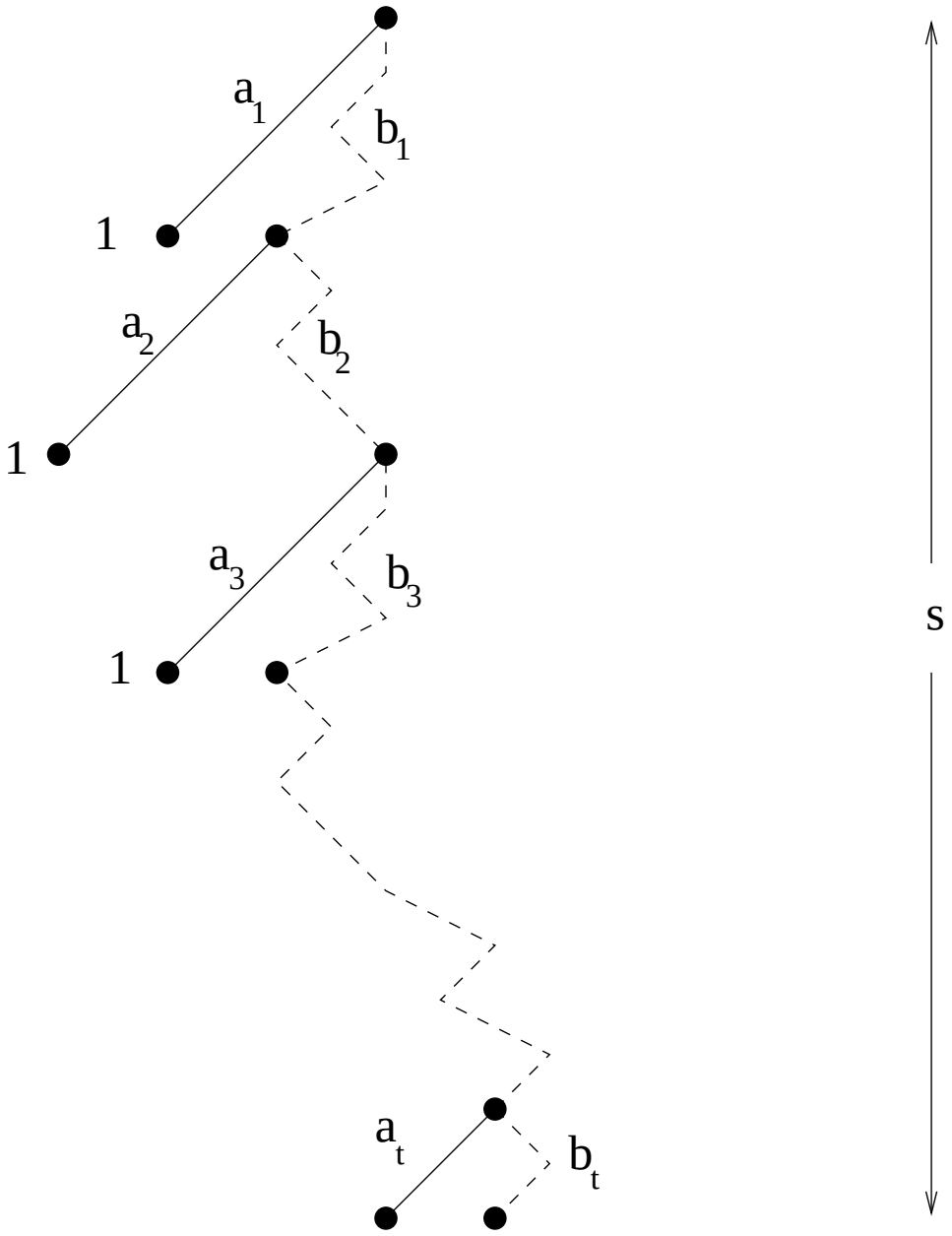

Figure 13.6: Decision tree $T(F|_\rho)$ with path of length s, $b = b_1 b_2 \cdots b_t$.

Assume that level one of the circuit — the nodes sitting above the inputs and their negations — consists of "or" gates. Thus, each of these gates g is a DNF formula whose maximum term size is one. By Lemma 13.2, with $p = 1/14, n_1 = n/14, k = 1$, we have,

$$|\{\rho \in \mathcal{R}_n^{n_1} \mid h(T(g|_\rho)) \geq \log s\}| \ < \ (2)^{-\log s} \cdot |\mathcal{R}_n^{n_1}| \ .$$

Since there are at most s gates at level one, the number of restrictions ρ such that $h(T(g|_\rho)) \geq \log s$ for some g is less than,

$$s \cdot (2)^{-\log s} \cdot |\mathcal{R}_n^{n_1}| \quad = \quad |\mathcal{R}_n^{n_1}| \ .$$

Thus, there is at least one restriction $\rho_1 \in \mathcal{R}_n^{n_1}$ under which all the gates at level one are CNF formulas with terms of size less than $\log s$. It follows that the "and" gates at level two are CNF formulas with terms of size less than $\log s$.

Let $g_2 = g|_{\rho_1}$ be any such gate. Using Lemma 13.2, with $k = \log s$, $p = 1/(14 \log s)$, $n_2 = n_1/(14 \log s)$, we have,

$$|\{\rho \in \mathcal{R}_{n_1}^{n_2} \mid h(T(g_2|_\rho)) \geq \log s\}| \ < \ (2)^{-\log s} \cdot |\mathcal{R}_{n_1}^{n_2}| \ .$$

Thus, there is a restriction $\rho_2 \in \mathcal{R}_{n_1}^{n_2}$ under which every gate at level two is a DNF formula all of whose terms have length less than $\log s$.

Repeating this argument through all d levels, we have a restriction $\rho = \rho_1 \rho_2 \cdots \rho_d \in \mathcal{R}_{n_d}^{n}$ such that the height $T(C|_\rho)$ of the decision tree of the root of the circuit is less than $\log s$. Observe that $n_d = n/(14^d (\log s)^{d-1})$. Let b be the restriction corresponding to any branch of the decision tree. It follows that $C|_{\rho b}$ is constant and has at least $r = n_d - (\log(s) - 1)$ inputs. □

Suppose that circuit C in Theorem 13.7 computes the parity of its n inputs. Then any restriction of C also computes the parity of its remaining inputs. Thus, if $1 \leq r$ in Theorem 13.7, then C must not compute PARITY. It follows that if C is a size s, depth d circuit computing parity on n inputs, then the following inequalities hold,

$$\begin{aligned} 1 &> n/(14^d (\log s)^{d-1}) - (\log(s) - 1) \\ \log s &> n/(14^d (\log s)^{d-1}) \\ (\log s)^d &> n/(14^d) \\ s &> 2^{\frac{1}{14} n^{\frac{1}{d}}} \ . \end{aligned}$$

We thus have the following lower bound on the number of iterations of a first-order quantifier block needed to compute PARITY. This corollary is optimal by Exercise 4.19.

We use the "big omega" notation for lower bounds. The "equation" $f(n) = \Omega(g(n))$ is equivalent to $g(n) = O(f(n))$. It means that for almost all values of n, $f(n)$ is at least some constant multiple of $g(n)$.

Corollary 13.8 *If* PARITY $\in$ FO$[s(n)]$, *then* $s(n) = \Omega(\log n / \log\log n)$, *and this holds even in the presence of arbitrary numeric predicates.*

Exercise 13.9 Show that PARITY is first-order reducible to REACH. Conclude that the same lower bound as in Corollary 13.8 holds for REACH. □

13.2 A Lower Bound for REACH_a

In this section, we prove a lower bound (Theorem 13.11) on the quantifier-rank needed to express the P-complete problem REACH_a (Definition 3.23), when ordering and the other numeric predicates are not available. If the same result were proved for the language with ordering, it would imply that NC is strictly contained in P, and in fact that $\bigcup_k$ DSPACE$[(\log n)^k]$ does not contain P.

Exercise 13.10 Show that REACH_a is expressible in FO-VAR(wo$\leq$)$[n, 2]$. [Hint: just write down the natural inductive definition of the alternating path relation.] □

In the remainder of this section, we prove the following lower bound:

Theorem 13.11 *Boolean query* REACH_a *is not expressible in quantifier rank* $2^{\sqrt{\log n}-1}$ *in the language without ordering.*

To prove Theorem 13.11, we construct graphs G_m and H_m with the following properties:

$$\begin{array}{ll} 1. & \|G_m\| = \|H_m\| < m^{1+\log m} \\ 2. & G_m \sim_m H_m \\ 3. & G_m \in \mathrm{REACH}_a;\ H_m \notin \mathrm{REACH}_a\ . \end{array} \tag{13.12}$$

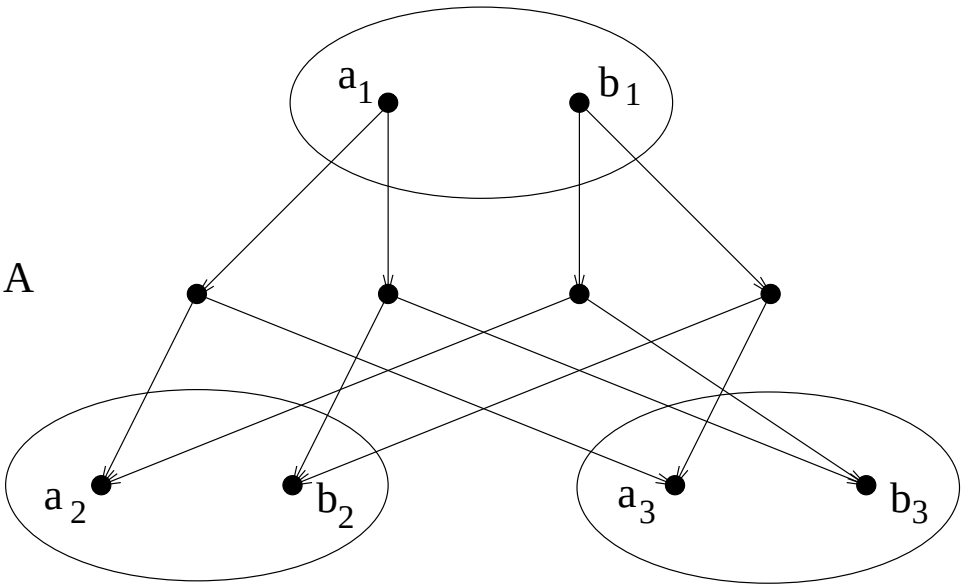

Figure 13.13: The Directed Switch X_d

Note that condition (1) implies that for $n = \|G_m\|$, $\log n < (1+\log m)(\log m)$, so $\sqrt{\log n} < 1 + \log m$, so $2^{\sqrt{\log n}-1} < m$. Thus, Equation (13.12) implies Theorem 13.11.

The first step in producing G_m and H_m is to introduce the building block out of which they will be constructed.

Lemma 13.14 *Let X_d be the alternating graph pictured in Figure 13.13. Then X_d has automorphisms that switch any two of the pairs $(a_1, b_1), (a_2, b_2)$, and (a_3, b_3), leaving the other pair fixed.*

Proof The idea is that when X_d is placed in a graph, each of the pairs will consist of one point that can reach d and one point that cannot. Note that the four points at the middle of X_d are "and"-nodes and the other points are "or"-nodes. The boolean formulas corresponding to alternating graph X_d are the following:

$$\begin{aligned} a_1 &\equiv (a_2 \wedge a_3) \vee (b_2 \wedge b_3) , \\ b_1 &\equiv (a_2 \wedge b_3) \vee (b_2 \wedge a_3) . \end{aligned}$$

The proof of the lemma is an easy computation. □

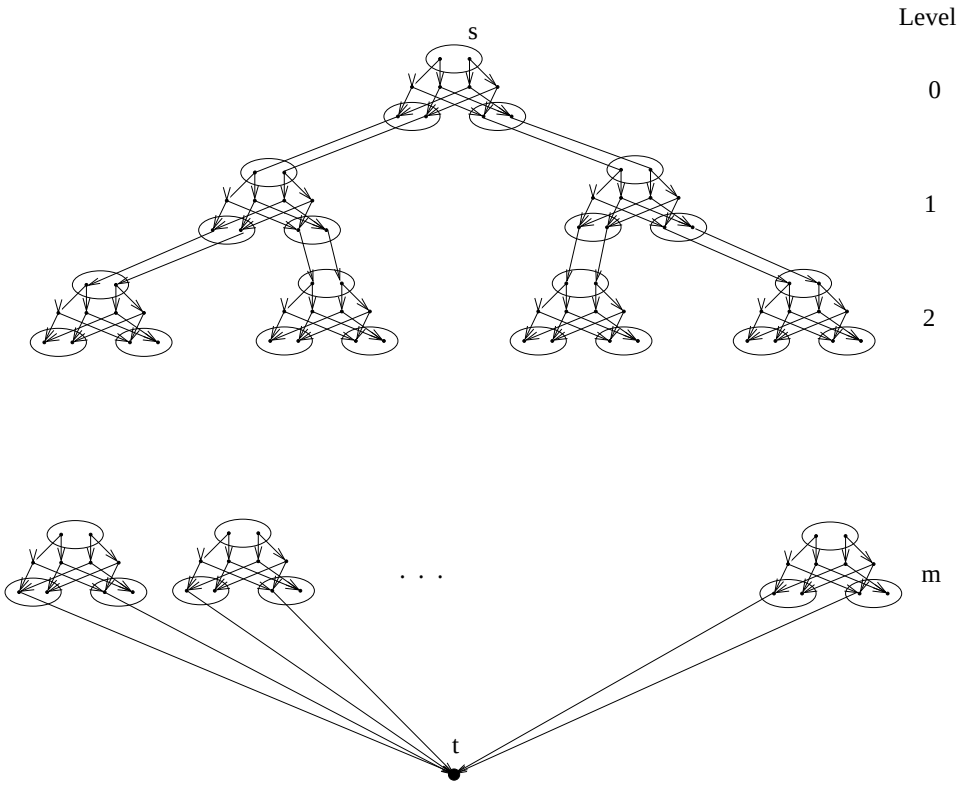

Figure 13.15: The graph $\tilde{D}_d(T_m)$.

Before we construct the graphs G_m and H_m satisfying Equation (13.12), we build exponential size graphs as a warm-up.

Let T_m be a complete binary tree of height m, with root r and edges directed from root to leaves. Define $X_d(T_m)$ to be the graph obtained by replacing each vertex v from T_m by a copy $X_d(v)$ of the switch X_d. Let y and z be the left and right children of v, respectively. Then $X_d(T_m)$ contains the edges $(a_2(v), a_1(y)), (b_2(v), b_1(y))$ and $(a_3(v), a_1(z)), (b_3(v), b_1(z))$. Furthermore, add an additional vertex t and draw the edges $(a_2(\ell), t)$ and $(a_3(\ell), t)$ for each leaf ℓ of T_m. Finally, interpret constant symbol s as $a_1(r)$. Define $\tilde{X}_d(T_m)$ to be the same as $X_d(T_m)$ except that s is interpreted as $b_1(r)$. Thus $X_d(T_m) \in$ REACH$_a$, but $\tilde{X}_d(T_m) \notin$ REACH$_a$. See Figure 13.15 for a diagram of $\tilde{X}_d(T_m)$.

The following observation about $X_d(T_m)$ and $\tilde{X}_d(T_m)$ leads to $\mathcal{D}$'s winning strategy in $\mathcal{G}_m(X_d(T_m), \tilde{X}_d(T_m))$:

Observation 13.16 *Suppose that in $\tilde{X}_d(T_m)$, we take any pair of edges, $(a_i(v), a_i(w))$, $(b_i(v), b_i(w))$, and* switch them*, i.e., replace them by $(a_i(v), b_i(w))$, $(b_i(v), a_i(w))$. Then the resulting graph is isomorphic to $X_d(T_m)$. In fact, switching any odd number of edge pairs in $X_d(T_m)$ ($\tilde{X}_d(T_m)$) yields a graph that is isomorphic to $\tilde{X}_d(T_m)$ ($X_d(T_m)$).*

Proof The proof follows from Lemma 13.14. First, take any single edge-pair-switch in $\tilde{X}_d(T_m)$. By Lemma 13.14 there is an automorphism of the graph that flips the pairs $a_i(v), b_i(v)$ and $a_1(v), b_1(v)$. The result thus pushes the edge switch up one level in the tree. When the top is reached, $a_1(r)$ and $b_1(r)$ have been switched i.e., $\tilde{X}_d(T_m)$ has been changed to $X_d(T_m)$. If there is more than one pair of switched edges, then in this way they can be pushed to the root one by one. Each time $\tilde{X}_d(T_m)$ is changed to $X_d(T_m)$ or vice-versa. □

Lemma 13.17 *For $m = 1, 2, \ldots,$ $X_d(T_m) \sim_m \tilde{X}_d(T_m)$.*

Proof By induction on m. This is clear for $m = 0$. Assume the lemma for m and consider the game $\mathcal{G}_{m+1}(X_d(T_{m+1}), \tilde{X}_d(T_{m+1}))$. Suppose that Samson's first move is to place a pebble on a vertex a in $X_d(v)$ for some $v \in T_m$. It does not matter whether a is in $X_d(T_{m+1})$ or $\tilde{X}_d(T_{m+1})$. Either $v = r$ is the root of T_{m+1} or it is in the left or right subtree of r. If v is in the left subtree, then Delilah should answer according to the isomorphism σ provided by Observation 13.16 between $X_d(T_{m+1})$ and $\tilde{X}_d(T_{m+1})$ with the edge pair $(a_3(r), a_1(w)), (b_3(r), b_3(w))$ switched, where w is the right child of r. Notice that w is now the root of a copy of $\tilde{X}_d(T_m)$. Any further moves in the right subtree should be answered according to Delilah's inductive winning strategy in the game $\mathcal{G}_m(X_d(T_m), \tilde{X}_d(T_m))$. Any further moves in the other part of the tree should be answered by the isomorphism σ. Thus, this strategy is always a win for Delilah. If the first move was in the right subtree, then Delilah's answer is similar. If v is the root of T_{m+1}, then Delilah may arbitrarily place the imaginary edge switch in the right subtree and answer according to the isomorphism σ. □

Exercise 13.18 Show that Samson wins the game $\mathcal{G}^3_{m+2}(X_d(T_m), \tilde{X}_d(T_m))$.

[Hint: Suppose that Samson places his first two pebbles on $a_1(v_1)$ and $a_2(v_2)$, where v_1 and v_2 are the children of the root of T_m. Delilah cannot answer with $a_1(v)$ and $a_2(v)$ on the other side or she will lose in two more moves. Thus, in two moves and three pebbles, Samson can push the difference between the two graphs one level down the tree.] □

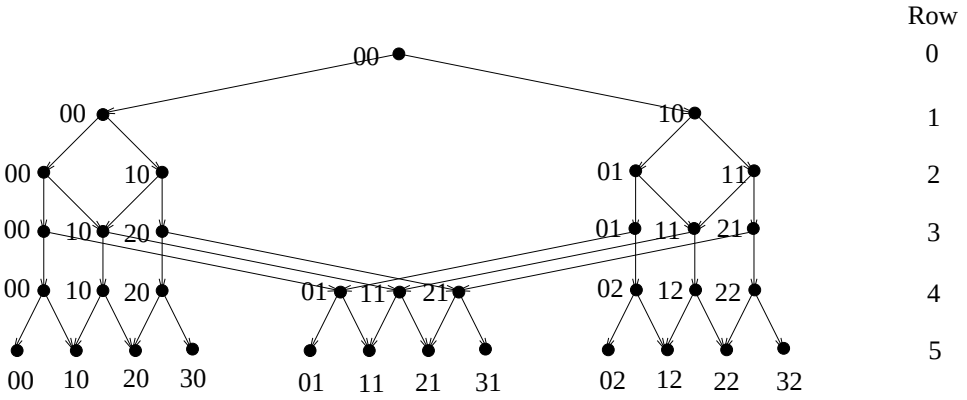

Figure 13.19: The graph D_2.

We continue with the proof of Theorem 13.11. The last step of the proof is the introduction of graphs D_k. $D_{\log m}$ will replace the binary tree T_m in the construction. $D_{\log m}$ has about $m^{\log m}$ vertices but it has essentially $\log m$ degrees of freedom — enough to let Delilah win the m-move game.

The graph $D_k = (V_k, E_k)$ is defined as follows. See Figure 13.19 for the graph D_2. The vertices V_k consist of $k+1$-tuples, $\langle x_1, \ldots, x_k, r\rangle$ where $x_1, \ldots, x_k$ can be thought of as coordinates and r is the row number. As we move from each row to the one below, one of the coordinates is expanded by one. Call a *block* of D_k, k consecutive rows. So from the top of one block to the top of the next, each coordinate is expanded by one.

$$
\begin{aligned}
V_k &= \left\{\langle x_1, \ldots, x_k, r\rangle \;\middle|\; r = ak + j, a < 2^k, 0 \le x_i \le a+1, x_i \le a \text{ for } i > j\right\} \\
E_k &= \left\{(\langle \bar{x}, r-1\rangle, \langle \bar{x}, r\rangle) \;\middle|\; \text{for all } r < k2^k\right\} \cup \\
&\quad \left\{(\langle x_1, \ldots, x_k, r-1\rangle, \langle x_1, \ldots x_i, x_i+1, x_{i+1} \ldots x_k, r\rangle) \;\middle|\; i \equiv r \,(\mathrm{mod}\, k)\right\}
\end{aligned}
$$

Let $G_m = X_d(D_{\log m})$ and $H_m = \tilde{X}_d(D_{\log m})$. See Figure 13.20 for a drawing of part of $X_d(D_2)$. Notice that in the directed tree, all internal nodes have in-degree one, but in D_k, some vertices have in-degree two. For such vertices v, there are several incoming edges to $a_1(v)$ and $b_1(v)$.

We show that the three conditions of Equation 13.12 hold. Conditions 1. and 3. are immediate. We must show that Delilah wins the m-move game on $X_d(D_{\log m})$ and $\tilde{X}_d(D_{\log m})$.

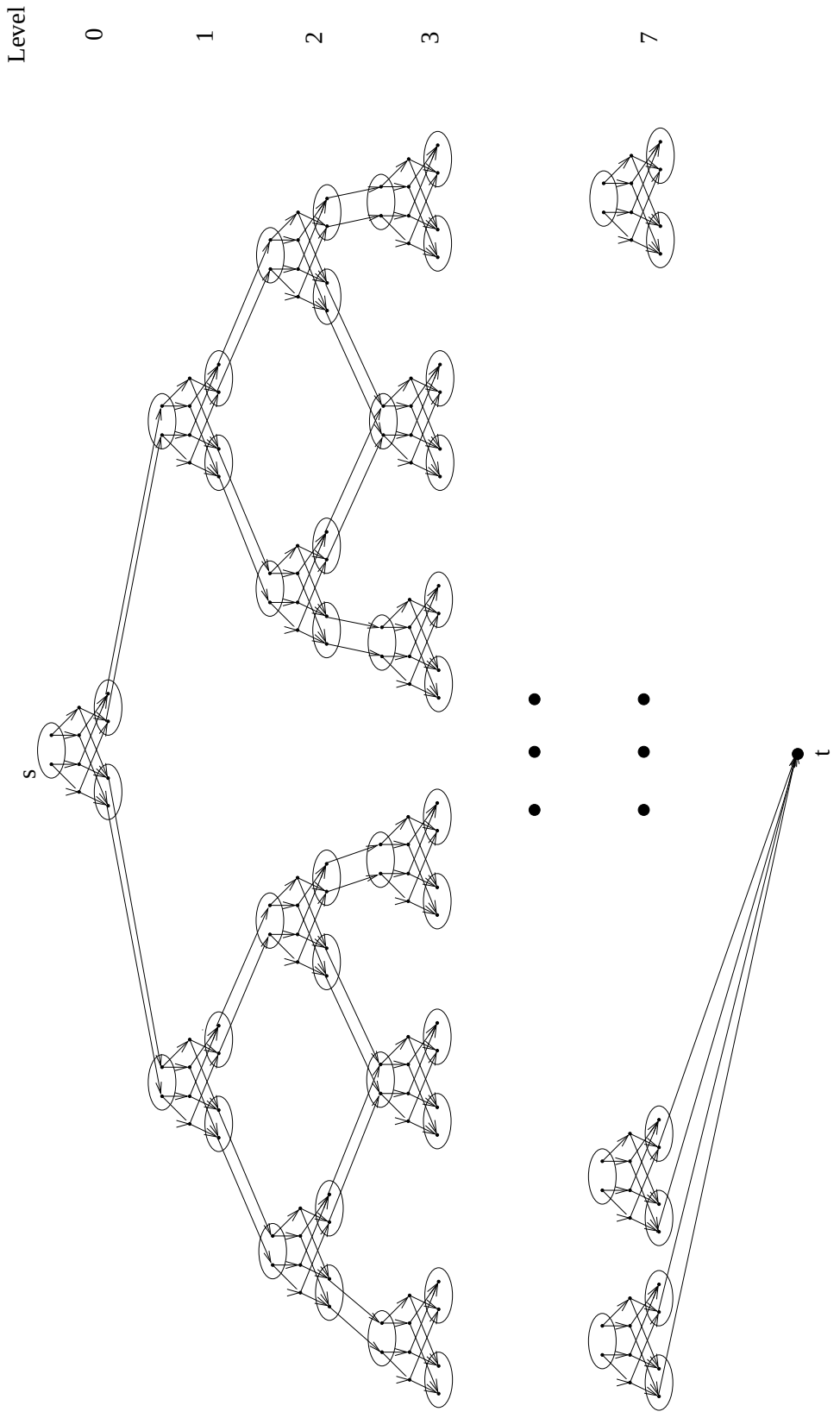

Figure 13.20: The graph $X_d(D_2)$.

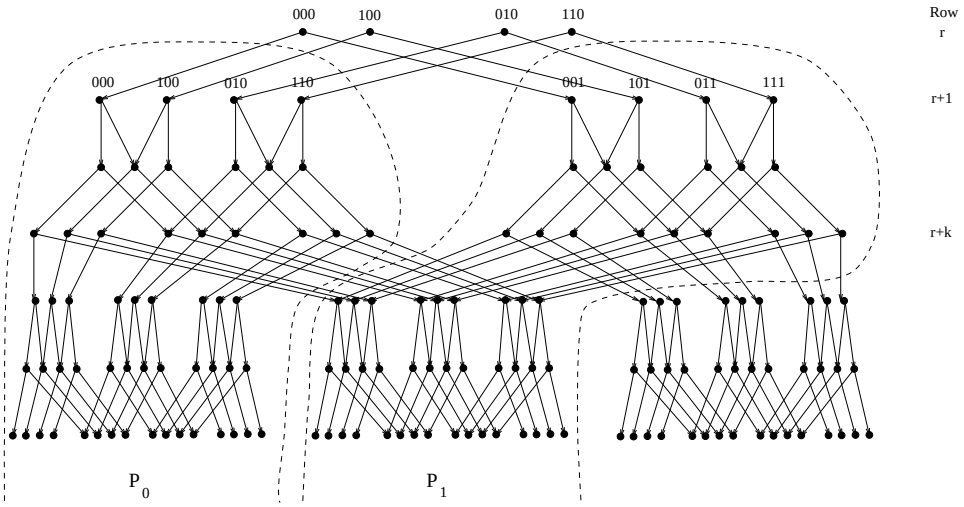

Figure 13.21: Proof of Claim 13.22, $k = 3, r = 2, i = 3$.

Think of a round of the game as labeling a vertex v in $D_{\log m}$: it is labeled "0" if Delilah answers with the same point as Samson, e.g., they both choose $a_1(v)$. It is labeled "1" if Delilah answers with the opposite point. The *labeling rule* is that if the two children of v are labeled, then v must be labeled with the "exclusive or" of its children's labels. Delilah wins the game as long as she never breaks the labeling rule and never labels a bottom vertex "1" or the root "0".

The crucial property of D_k is stated in the following claim. It says that it does Samson no good in the 2^k-move game, to choose a vertex more than k levels below where he has forced a vertex to be labeled "1". Thus, in m moves, Samson cannot force a vertex at the bottom level to be labeled "1".

Claim 13.22 *Suppose row r of D_k is entirely labeled and let any $2^k - 1$ vertices on or below row $r + k$ be chosen. If the chosen vertices are all labeled "0", then there is still a labeling of the rest of D_k that is consistent with row r.*

Proof By induction on k. For $k = 1$, let v be the chosen vertex on or below row $r + 1$. Suppose v is on row $r + 1$ of D_1. Let ℓ be a labeling of row $r + 1$ that generates the required labeling of row r. Observe that $\bar{\ell}$ — the complement of ℓ — generates the same labeling of row r. Clearly one of ℓ and $\bar{\ell}$ labels v "0" as desired. If v is below row $r + 1$, then take an arbitrary labeling of row $r + 1$ and proceed down to the row above v and then use the same argument.

Inductively, assume the claim is true for all $k' < k$. Let row r be fixed and suppose that $2^k - 1$ vertices of D_k have been chosen on or below row $r + k$. Let i be the coordinate that is expanded as we pass from row r to row $r + 1$. That is, $i \equiv r + 1 \pmod{k}$. Let t be the maximum coordinate i occurring in row $r + k$. Let $P_0, P_1, \ldots, P_t$ be the subsets of D_{k+1} below row r, projected onto values $0, 1, \ldots, t$ of coordinate i,

$$P_j \;=\; \left\{ \langle x_1, \ldots, x_{k+1}, s \rangle \;\middle|\; s > r, x_i = j \right\} .$$

(See Figure 13.21.)

Observe that each of the P_j's is a copy of D_{k-1} except that each row $k - 1$ is repeated. There can be at most one of the P_j's — call it P_{j_0} — that contain at least 2^{k-1} chosen vertices. Assume that all the vertices in P_{j_0} have been labeled "0". By induction, we can label row $r + 1$ of the rest of the P_j's as we please. Thus, we can label row r of D_k as we please. □

This completes the proof of Theorem 13.11. Observe that Delilah's winning strategy in the game $\mathcal{G}_m(X_d(D_{\log m}), \tilde{X}_d(D_{\log m}))$ is in fact a winning strategy in the bijection game, $\mathcal{G}^k_{B,m}(X_d(D_{\log m}), \tilde{X}_d(D_{\log m}))$ (Definition 12.22). At each move, for each vertex $v \in D_k$, Delilah decides whether she would label this vertex "0" or "1". In the former case, she maps every vertex in $X(v) \subset X_d(X_{\log m}))$ to the same vertex in $\tilde{X}_d(D_{\log m})$. In the latter case, she maps the vertices according to one of the automorphisms that switch $a_1(v)$ and $a_2(v)$ as given in Lemma 13.14.

Thus, we have proved,

Corollary 13.23 *Boolean query* REACH_a *is not expressible in quantifier rank* $2^{\sqrt{\log n}-1}$ *even in language* FO(COUNT).

For large n, the function $2^{\sqrt{\log n}}$ dominates $(\log n)^k$, for any value of k. Recall that class NC is equal to FO$[(\log n)^{O(1)}]$ (Corollary 5.26). Thus, if Theorem 13.11 went through with ordering, we would have proved that NC is strictly contained in P. This would mean that polynomial-time complete problems are *inherently sequential* in that they cannot be computed in parallel time $(\log n)^{O(1)}$ using polynomially many processors.

Of course, problem REACH_a is expressible in FO(wo$\leq$)(LFP). In the next section, we present a different use of switch X (Figure 13.13). We prove that the language FO(wo$\leq$)(LFP, COUNT) is strictly contained in order-independent P.

13.3 Lower Bound for Fixed Point and Counting

We now prove that the language with fixed point and counting, FO(wo$\leq$)(LFP, COUNT), falls far short of capturing order-independent polynomial-time.

The argument is similar to the lower bound on REACH_a (Theorem 13.11). Let switch X be the graph shown in Figure 13.24. This is the undirected version of switch X_d of Figure 13.13. Notice that the a vertices are circled in Figure 13.24 to distinguish them in the drawing from their companion b vertices. Note that the four central vertices have the property that each of them is connected to an even number of circled vertices. Just as in Lemma 13.14, we have that,

Lemma 13.25 *Let X be the graph pictured in Figure 13.24. Then X has automorphisms that switch any two of the pairs of a and b vertices, leaving the other pair fixed.*

Using switch X, we construct a sequence of pairs of graphs that are computationally simple to distinguish but require a linear number of variables to distinguish, even in the presence of counting:

Theorem 13.26 *There exists a sequence of pairs of graphs $\{A_n, \tilde{A}_n\}$, $n \in \mathbf{N}$, admitting a linear time canonical labeling algorithm and having the following additional properties:*

1. *A_n and $\tilde{A}_n$ have $O(n)$ vertices.*
2. *A_n and $\tilde{A}_n$ have degree three and color class size four.*
3. *$A_n \equiv_{\mathcal{C}^n} \tilde{A}_n$.*
4. *A_n is not isomorphic to $\tilde{A}_n$.*

Before we prove Theorem 13.26 note a few of its consequences:

Corollary 13.27 FO(wo$\leq$)(LFP, COUNT) *is strictly included in order-independent polynomial-time.*

Proof By Theorem 13.26, the problem of distinguishing A_n from $\tilde{A}_n$ is in order-independent P. Suppose that there were a sentence $\varphi \in$ FO(wo$\leq$)(LFP, COUNT) that distinguished A_n from $\tilde{A}_n$. Let k be the number of variables in φ.

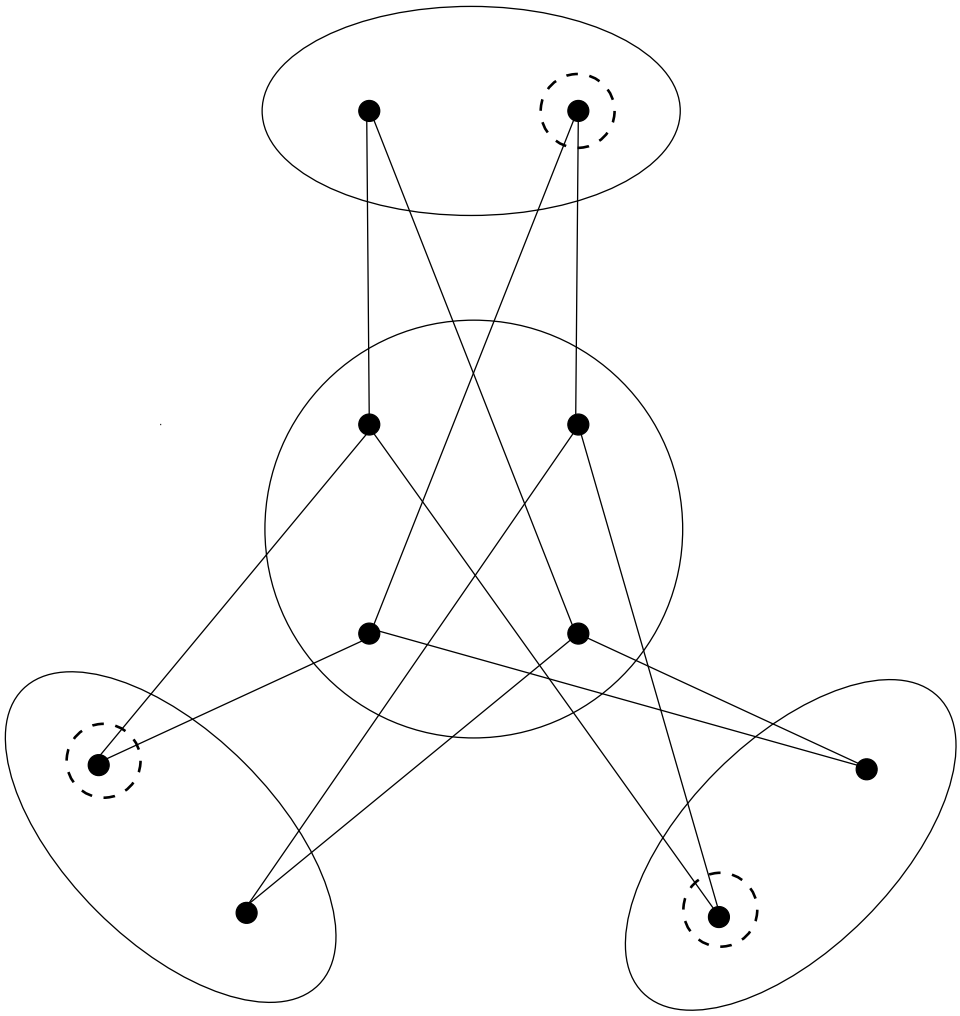

Figure 13.24: Switch X drawn with the a-vertices circled.

Thus there is a sentence $\varphi' \in \mathcal{C}^k$ such that $\langle A_n, \tilde{A}_n \rangle \models \varphi$ and $\langle A_n, A_n \rangle \models \neg\varphi$. However, we know that $A_n \sim_{\mathcal{C}^n} \tilde{A}_n$, sot $\langle A_n, A_n \rangle \sim_{\mathcal{C}^n} \langle A_n, \tilde{A}_n \rangle$ because Delilah's winning strategy in the first game carries over to the second by using her strategy in the second components and the identity map in the first components. This is a contradiction when $n \geq k$. □

The following corollary of Theorem 13.26 is in sharp contrast to Proposition 12.9, which says that three variables suffice to identify graphs of color class size three.

Corollary 13.28 *A linear number of variables is required to identify graphs of color class size 4, even in the presence of counting. In symbols, var*$(\mathrm{CC}_4) = \Omega(n)$ *and vc*$(\mathrm{CC}_4) = \Omega(n)$.

Proof of Theorem 13.26: Let G be an undirected graph that is regular of degree three. Define $X(G)$ to be the graph in which each vertex of G is replaced by $X(v)$, a copy of the switch X. For each edge (u, v) of G, a pair of vertices denoted by $a(u, v), b(u, v)$ is selected from $X(u)$, and similarly, the pair of vertices $a(v, u), b(v, u)$ is selected from $X(v)$. Edges $(a(u, v), a(v, u))$ and $(b(u, v), b(v, u))$ are drawn. See Figures 13.29 and 13.30 for a sample degree-three graph H and the corresponding $X(H)$.

If G has an ordering on its vertices, then $X(G)$ inherits a partial ordering. Call the four central vertices of $X(v)$, $c_i(v)$, $1 \leq i \leq 4$. Then vertices $a(u, v), b(u, v)$, and $c_i(v)$ are partially ordered according to the lexicographic ordering of $\langle u, v \rangle$, and $\langle v, v \rangle$. Observe that if G is ordered, then $X(G)$ has color class size 4 in the sense that the partial ordering distinguishes all vertices except the pairs $a(u, v), b(u, v)$ and the quadruples $c_1(v), \ldots, c_4(v)$.

From now on in this section, G will be a regular, degree-three graph with an ordering on its vertices. Define $\tilde{X}(G)$ to be the graph $X(G)$ except that the edges are flipped between $X(v_1)$ and $X(v_2)$, for (v_1, v_2) the lexicographically first edge in G. By "flipped" we mean that instead of the edges $(a(v_1, v_2), a(v_2, v_1))$, $(b(v_1, v_2), b(v_2, v_1))$, $\tilde{X}(G)$ has the edges, $(a(v_1, v_2), b(v_2, v_1))$, $(b(v_1, v_2), a(v_2, v_1))$.

Compare the drawing of $\tilde{X}(H)$ in Figure 13.31 with the drawing of $X(H)$ in Figure 13.30.

The following observation is similar to Observation 13.16,

Observation 13.32 *Let G be any regular, degree-three, connected graph. Let $\hat{X}(G)$ be like $X(G)$ except that exactly t pairs of edges are flipped. Then $\hat{X}(G)$ is isomorphic to $X(G)$ iff t is even and $\hat{X}(G)$ is isomorphic to $\tilde{X}(G)$ iff t is odd.*

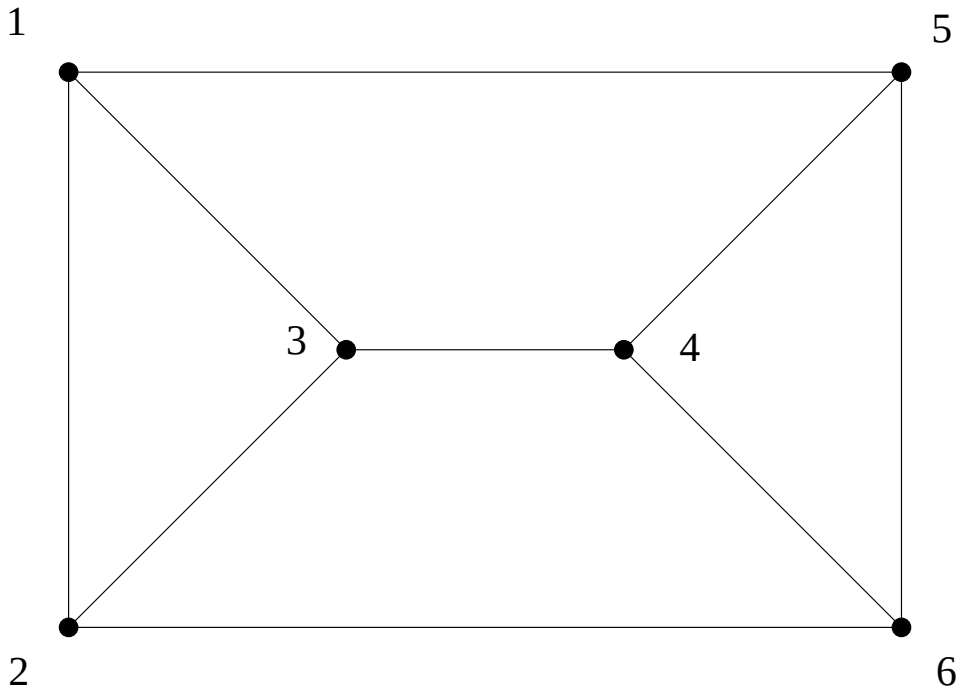

Figure 13.29: H is a regular, degree-three graph.

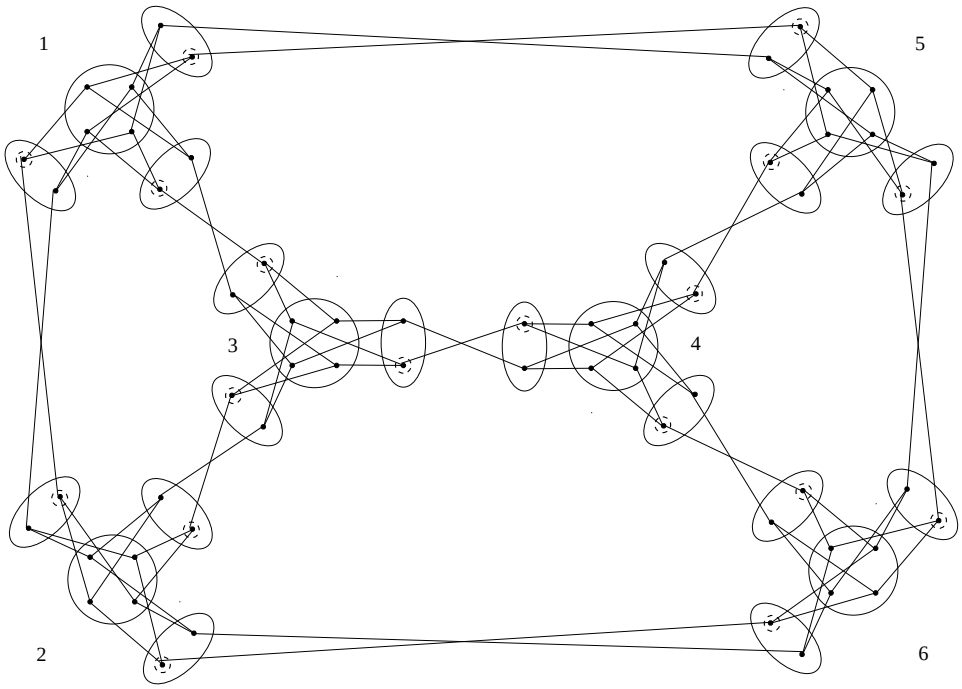

Figure 13.30: The graph $X(H)$.

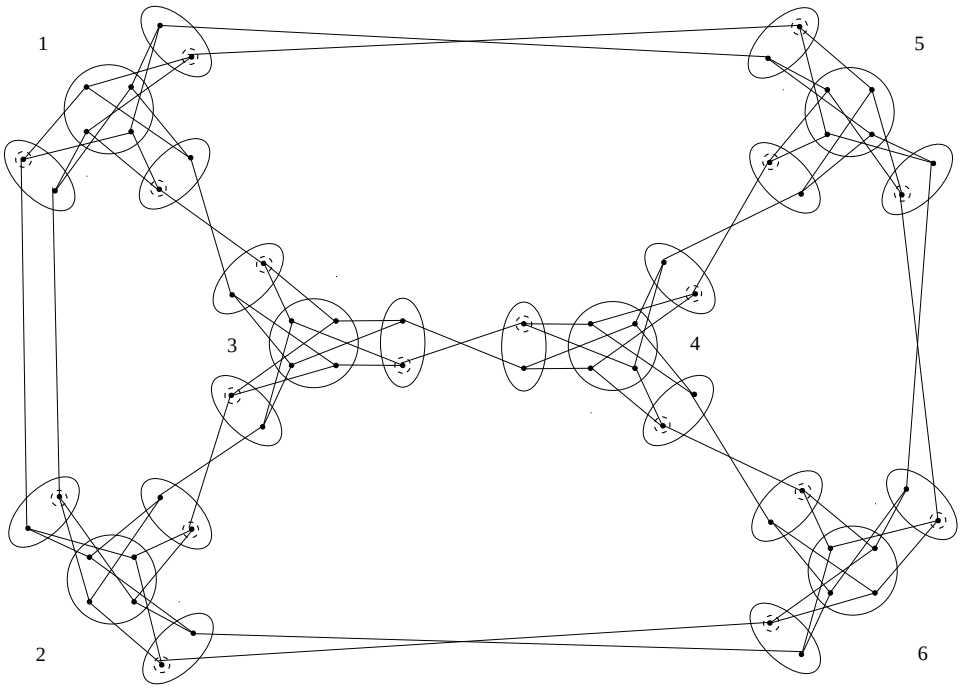

Figure 13.31: The graph $\tilde{X}(H)$.

The following are amusing and useful exercises.

Exercise 13.33 Prove Observation 13.32. The main subtlety is in proving that $X(G)$ is not isomorphic to $\tilde{X}(G)$. □

Let STRAIGHT be the set of graphs Z such that Z is equal to $X(G)$ for some ordered, regular, degree-three graph G. Similarly, let FLIP be the set of such graphs equal to $\tilde{X}(G)$ for such a G. In the next exercise you are asked to show that mutually exclusive boolean queries STRAIGHT and FLIP are each computable in linear time. Let $\oplus$ be the counting-mod-2 quantifier:

$$(\oplus x)\varphi(x) \qquad \equiv \qquad \text{“there are an odd number of } x\text{'s satisfying } \varphi.$$

In fact, STRAIGHT and FLIP are expressible — over ordered graphs — in FO($\oplus$). The class FO($\oplus$) is equivalent to circuit class AC^0 extended by counting-mod-two gates in addition to the usual “and”, “or” and “not” gates. This very small complexity class is strictly contained in $\text{ThC}^0 = \text{FO}(M)$ (Fact 13.37).

Exercise 13.34 Prove that STRAIGHT and FLIP are computable in linear time on a RAM. Prove also that they are expressible in the language FO($\oplus$).

[Hint: if Z is equal to $X(G)$ or $\tilde{X}(G)$ for an ordered graph G, then its ordering relation groups together each pair $a(u,v), b(u,v)$. Using the ordering of Z, we can label the first element in this pair a and the second b. This is the key point: the ordering gives us a global labeling of all these pairs. Now, each pair of edges between a,b-pairs is *straight* if the edge is from a to a and b to b and *flipped* otherwise. Similarly, each $X(v)$ is straight if the c_i's are each connected to an even number of a's and flipped if they are each connected to an odd number of a's. (If neither, then Z is not in FLIP or STRAIGHT.) The algorithm has only to count the number of flips mod two.] □

Let $G = (V, E)$ be a connected graph. Define a *separator* of G to be a subset $S \subset V$ such that the induced subgraph on $V - S$ has no connected component with more than $|V|/2$ vertices. A probabilistic construction shows that there exist regular, degree-three graphs whose separators all have size $\Omega(n)$ [Ajt87].

Let $T_1, T_2, \ldots$ be a sequence of regular, degree-three graphs such that T_n has $O(n)$ vertices and its smallest separator has size at least $n + 1$. The fact that T_n has only large separators means that it is “very connected”. For this reason, the flip in $\tilde{X}(T_n)$ can be almost anywhere. Even after we have pinned down a set of n vertices, S, the largest connected component of $T_n - S$ still includes over half the vertices of T_n. The flip can hide inside this largest connected component and never

be pinned down by Samson if he has only n pebbles. The following is the key idea in the proof of Theorem 13.26.

Lemma 13.35 *As above, let T_k be a regular, degree-three graph whose smallest separator has size at least $k+1$. Then,*

$$X(T_k) \sim_C^k \tilde{X}(T_k) .$$

Proof We show that Delilah wins bijection game $\mathcal{G}_B^k(X(T_k)\tilde{X}(T_k))$ (Definition 12.22).

We know by Observation 13.32 that if we flip any edge pair in $X(T_k)$, then the resulting graph is isomorphic to $\tilde{X}(T_k)$. After move r, let Q_r be the largest connected component in $T_k - P_r$, where P_r is the set of all vertices $v \in T_k$ such that just after move r, some pebble is placed on a vertex in $X(v)$. Since T_k has no separator of size less than $k+1$, Q_r contains over half of the vertices of T_k.

Delilah's winning strategy is to maintain the following invariant,

$$\begin{array}{l}\text{For each vertex } v \in Q_r, \text{ let } X^v(T_k) \text{ be } X(T_k) \\ \text{with an edge pair adjacent to } X(v) \text{ flipped.} \\ \text{There is an isomorphism } \eta_{r,v} \text{ from } X^v(T_k) \text{ to } \tilde{X}(T_k), \text{ such that} \\ \text{for all } x_i \in \text{dom}(\alpha_r),\ \eta_{r,v}(\alpha_r(x_1)) = \beta_r(x_i) .\end{array} \tag{13.36}$$

Clearly Invariant 13.36 holds before the first move. Suppose that it holds just after move r, and in move $r+1$, let Samson pick up pebble pair i. Delilah responds with the bijection ρ_{r+1} defined as follows:

For each $v \in T_k$, let S_v consist of v together with all vertices w from T_k such that a pebble — not including x_i — is currently on a vertex in $X(w)$. Let C_v be the largest connected component of $T_k - S_v$. Since S_v consists of at most k vertices, C_v contains more than half of the vertices in T_k as does Q_r. Let z be a vertex in $Q_r \cap C_v$. Define ρ_{r+1} to be equal to $\eta_{r,z}$ on $X(v)$.

At the end of move $r+1$, Samson places the x_i pebbles on a vertex g in some $X(v)$, and $\rho_{r+1}(g) = \eta_{r,z}(g)$. Delilah has not lost because $\eta_{r,z}$ is an isomorphism that maps the currently pebbled points to pebbled points.

After move $r+1$, the new Q_{r+1} is equal to S_v. Define $\eta_{r+1,z} = \eta_{r,z}$. For all other vertices $w \in Q_{r+1}$, consider a path from z to w that stays within Q_{r+1}. Define $\eta_{r+1,w}$ to be the modification of $\eta_{r,z}$ that arises by pushing the flip from z to w. This affects no points outside Q_{r+1}.

Thus, Invariant 13.36 holds after move $r+1$. □

Finally, choosing $A_n = X(T_n)$, and $\tilde{A}_n = \tilde{X}(T_n)$ we have proved Theorem 13.26. □

Historical Notes and Suggestions for Further Reading

Theorem 13.1 was originally proved by Ajtai [Ajt83] and independently by Furst, Saxe, and Sipser [FSS84]. The paper [Ajt83] is very rich. It also proves, among other things, that there is a strict arity hierarchy in SO$\exists$. Sipser also proved that there is a strict depth hierarchy for AC^0, [Sip83]. That is, there are first-order boolean queries of alternation depth $k+1$ that are not in non-uniform, alternation depth k first-order.

Yao improved the bounds of Theorem 13.1, showing that a bounded depth AC^0 type circuit must have exponential size to express parity [Yao85]. Håstad simplified Yao's proof and strengthened the bounds, thus proving Lemma 13.2 and Corollary 13.8 [Has86].

Finally, Razborov used an algebraic argument to show that PARITY $\notin$ FO$(\oplus_3)$, where $\oplus_3$ is the counting mod 3 quantifier [Raz87]. This was extended by Smolensky to show the following [Smo87]:

Fact 13.37 *For distinct primes p and q,* FO$(\oplus_p)$ *is not contained in* FO$(\oplus_q)$*, and these are thus both strictly contained in* FO(M) = ThC^0.

Theorem 13.26 was proved by Cai, Fürer and Immerman [CFI92]. It provides an $\Omega(n)$ lower bound on the number of variables needed to identify graphs on n vertices (Corollary 13.28). Before this theorem was proved, it was believed that a constant number of variables might suffice. This would have led to a polynomial-time algorithm for general graph isomorphism. There had been considerable work in this direction by Weisfeiler, Lehman, and others [Wei76].

Hella introduced the Bijection Game (Definition 12.22) and proved that it is equivalent to the counting game (Theorem 12.23). He used this result to show that the lower bound of Theorem 13.26 proves something much stronger: adding unary generalized quantifiers to FO(LFP, COUNT) never captures order-independent P [He96].

There are many other descriptive lower bounds without ordering that are worth noting. Grädel and McColm proved lower bounds on logics with transitive closure

operators [Grä92a, GM96, GM95]. Etessami and Immerman proved upper and lower bounds on transitive closure logics concerning tree isomorphism and canonization [EI95a]. This was part of Etessami's thesis on the power of local orderings [Ete95a, Ete95].

Grohe proved the following very strong theorem that there is a strict arity hierarchy for transitive closure, least fixed point, and partial fixed point logics [Gro96a]:

Fact 13.38 (Arity Hierarchy Theorem) *For every $k > 1$, there is a formula $\varphi_k(x_1, \ldots, x_k, x'_1, \ldots, x'_k)$ that is expressible in* FO(TC(arity k)) *but is not expressible in* FO(PFP(arity $k-1$))*, and thus not in* FO(LFP(arity $k-1$)) *or* FO(TC(arity $k-1$)).