SIAM J. DISCRETE MATH.
Vol. 14, No. 4, pp. 471–480
© 2001 Society for Industrial and Applied Mathematics

CLASSIFICATION OF HOMOMORPHISMS TO ORIENTED CYCLES AND OF k-PARTITE SATISFIABILITY*

TOMÁS FEDER†

Abstract. We show that, for every choice of an oriented cycle H, the problem of whether an input digraph G has a homomorphism to H is either polynomially solvable or NP-complete. Along the way, we obtain simpler proofs for two known polynomial cases, namely, oriented paths and unbalanced oriented cycles, and exhibit two new simple polynomial cases of balanced oriented cycles. The more difficult cases of the classification are handled by means of a new problem, the bipartite boolean satisfiability problem. In general, the k-partite boolean satisfiability problems are shown to be either polynomially solvable or NP-complete, thus generalizing Schaefer's classification of boolean satisfiability problems.

Key words. digraph homomorphism, oriented cycles, constraint satisfaction

AMS subject classifications. 05C99, 05C20, 05C38

PII. S0895480199383353

1. Introduction.

A large class of problems in artificial intelligence and other areas of computer science can be viewed as *constraint-satisfaction problems*. This includes problems in machine vision, belief maintenance, scheduling, temporal reasoning, graph theory, and satisfiability. An instance of constraint-satisfaction is given by a pair I, T of finite relational structures over the same vocabulary. (The vocabulary is the list of relation names and their arities.) The instance is satisfied if there is a homomorphism from I to T; that is, there exists a mapping h such that for every tuple $(x_1, \ldots, x_k) \in R_i$ in I, we have $(h(x_1), \ldots, h(x_k)) \in R_i$ in T. Intuitively, the elements of I should be thought of as variables, and the elements of T should be thought of as possible values for the variables. The tuples in the relations of I and T should be viewed as constraints on the set of allowed assignments of values to variables. The set of allowed assignments is nonempty iff there exists a homomorphism from I to T.

It is well known that the constraint-satisfaction problem is NP-complete. In practice, however, one often encounters the situation where the structure T (which we call the *template*) is fixed, and it is only the structure I (which we call the *instance*) that varies.

Feder and Vardi [1] considered constraint-satisfaction with respect to fixed templates and asked whether each template defines either a polynomially solvable problem or an NP-complete problem. In other words, they asked whether there is a *dichotomy*; i.e., there is no constraint-satisfaction problem with respect to a fixed template that is not polynomially solvable or NP-complete. Such dichotomies have been shown in several special cases of constraint-satisfaction; see [1] for an overview. They showed that the class of constraint-satisfaction problems with respect to fixed templates contains as special cases the problems of k-satisfiability, k-colorability, systems of linear equations modulo q, and labeled graph isomorphism; they also showed that the general case is equivalent to the case of digraphs, known as digraph homomorphism.

That is, let H be a fixed digraph. For an input digraph G, the problem is to decide whether there is a homomorphism from G to H, i.e., a mapping from the vertices of G

*Received by the editors February 1, 1999; accepted for publication (in revised form) August 6, 2001; published electronically October 4, 2001.
http://www.siam.org/journals/sidma/14-4/38335.html
†268 Waverley St., Palo Alto, CA 94301 (tomas@theory.stanford.edu).

to vertices of H such that adjacent vertices in G map to adjacent vertices in H in the same direction. As said above, classifying the digraphs H as defining polynomially solvable or NP-complete problems would yield such a classification on templates for constraint-satisfaction.

The digraph homomorphism problem has been extensively studied; see, e.g., [2, 3, 7, 8, 9]. One of the cases that they focused on is that of oriented cycles, i.e., the case where H is a cycle with each of its edges oriented in one of the two possible directions. Partial results indicating that some cycles define polynomial [9] and some define NP-complete [2] problems were obtained. In particular, the cases of oriented paths and of unbalanced oriented cycles were classified as polynomially solvable. In this paper, we complete the classification by showing that each oriented cycle H gives rise to either a polynomially solvable or an NP-complete digraph homomorphism problem. The approach is based on ideas of Feder and Vardi [1], which suggest several classes of polynomially solvable constraint-satisfaction problems, and uses explicitly the classification for the boolean case of constraint-satisfaction by Schaefer [10].

What is needed is in fact a generalization of the boolean constraint-satisfaction classification by Schaefer, namely, one where the template does not contain just two elements, yet every element of the input structure is restricted in the instance to map to one of two specific elements. That is, every element ranges over a set $S_i = \{0_i, 1_i\}$ for k different sets S_i (that is, what we call the k-partite boolean satisfiability problem), and we classify such problems as polynomially solvable or NP-complete.

2. Some polynomial cases. Feder and Vardi [1] observed that all known polynomially solvable cases of constraint satisfaction can be solved by means of Datalog and group theory, defining the bounded width and subgroup classes of problems, respectively. Within the bounded width class, they identified several subclasses for which membership is decidable: in particular, the width 1 class (also known as the tree duality class [5], with its extension the extended tree duality class [1]), and the bounded strict width class. This last class, the bounded strict width class, will be of special interest here.

The bounded width class uses a canonical algorithm, for problems of width (l, k), that involves inferring all possible constraints on l variables at a time by considering k variables at a time. We may in addition require that if this inference process does not reach a contradiction (the empty set), then it should be possible to obtain a solution by greedily assigning values to the variables one at a time while satisfying the inferred 1-constraints. We say that a constraint-satisfaction problem that can be solved in this way has *strict width* (l, k) and say that it has *strict width* l if it has strict width (l, k) for some k. It turns out that strict width l is equivalent to strict width (l, k), for all $k > 1$, so we can assume $k = l + 1$.

This intuition behind strict width l can also be captured in two other ways. First, we can require that if we have an instance and after assigning specific values to some of the variables we obtain an instance with no solution, then some l out of the specific value assignments chosen are sufficient to give an instance with no solution. We refer to this property as the l-Helly property. Second, we could require that there exists a function g that maps $l + 1$ elements from the domain of the structure T to another element with the property that if all but at most one of the $l + 1$ arguments are equal to some value b, then the value of g is also b. Furthermore, for all relations R in the template, if we have $l + 1$ tuples satisfying R, then the tuple obtained by applying g componentwise also satisfies R. We call this property the l-mapping property. The following is from Feder and Vardi [1].

THEOREM 1. *Strict width l, the l-Helly property, and the l-mapping property are equivalent.*

We shall show here that several cases of digraph homomorphism have strict width 2, by using the 2-mapping property, and are therefore polynomially solvable. That is, fix a digraph H. Then the digraph homomorphism problem for H has strict width 2 iff there is a mapping $t = g(x, y, z)$ from triples of vertices in H to vertices in H such that $g(x, x, x) = g(x, x, y) = g(x, y, x) = g(y, x, x) = x$, and if (x, x'), (y, y'), (z, z') are edges of H, then $(t, t') = (g(x, y, z), g(x', y', z'))$ is also an edge of H.

Oriented paths were shown to be polynomially solvable in [2, 3, 8]. Here we show this by establishing strict width 2. This stronger result will be of use later in connection to the 2-Helly property.

THEOREM 2. *Oriented paths have strict width* 2.

Proof. Oriented paths have vertices numbered $1, 2, \ldots, r$, with all edges from i to $i + 1$, or from $i + 1$ to i, and exactly one of both for each i. Let $g(x, y, z) =$ median (x, y, z), i.e., the middle value out of x, y, z. Note that two edges (x, x') and (y, y') with $x < y$ give $x' \leq y'$, so for the definition of the 2-mapping property, the order of vertices is preserved by the adjacency relation, and hence the same argument is selected by g before and after traversing an edge; that is, the 2-mapping property holds. □

An oriented cycle is *balanced* if it has the same number of edges on the cycle in one direction and in the other, *unbalanced* otherwise. It is known [2, 12] that unbalanced cycles are polynomially solvable. Here we show this by establishing strict width 2.

THEOREM 3. *Unbalanced cycles have strict width* 2.

Proof. Traverse the cycle in one direction, obtaining vertices $0, 1, \ldots, k$, where vertices 0 and k designate the same vertex. Define level(i) in such a way that if $(i, i + 1)$ is an edge, then level($i + 1$) = level(i) + 1; otherwise, if $(i + 1, i)$ is an edge, then level($i + 1$) = level(i) − 1. Notice that level (k) ≠ level(0), since the oriented cycle is unbalanced. Assume level(k) < level(0) (by an arbitrary choice of direction) and also that the smallest level is 0. The levels thus look in order like this: $l + m, \ldots, 0, \ldots, 0, \ldots, 0, \ldots, l$ for some $m > 0$. Adding m to the levels starting from the first 0, we obtain $l + m, \ldots, 0 \equiv m, \ldots, m, \ldots, m, \ldots, l + m$; i.e., by again choosing the starting vertex of the ordering, we have $m, \ldots, 0$ with a single 0.

Now that we have ensured that level(0) = m, level(k) = 0, and level(i) > 0 for $i \neq k$, define the lexicographic ordering on pairs (level(i), i). We refer to the sets of vertices whose levels differ by a multiple of m as *layers*. To define the 2-mapping g, if the three arguments are in three different layers, we always return the first argument; if only two of the three arguments are in the same layer, we always return the first of these two arguments; the only interesting case thus has all three arguments in the same layer. For the layer containing the endpoint $k \equiv 0$, the two lexicographically smallest pairs are precisely the two corresponding $(0, k)$ and $(m, 0)$. Also note that if (x, x') and (y, y') are edges with $x < y$ in the lexicographic ordering, and with x and y in the same layer, then x' and y' are in the same layer, and $x' \leq y'$ in the lexicographic ordering. We can therefore define $g(x, y, z) =$ median(x, y, z) in the lexicographic ordering, as for paths in the previous theorem, with x, y, z in the same layer, thus obtaining the 2-mapping property. □

We now turn to balanced cycles. For balanced cycles, the above procedure for assigning levels from the previous theorem assigns the same level to vertices 0 and k, i.e., $m = 0$. Denote the lowest and highest levels by l and h. Then, as we traverse the cycle, we encounter a sequence of l's and h's, which is in general of the form $(l^+h^+)^i$

with $i \geq 1$. Here l^+h^+ denotes one or more repetitions of l's followed by one or more repetitions of h's, and i indicates the number of times this pattern is encountered. We first consider the case $i = 1$ and show that it has strict width 2. We next consider the case $i = 2$ and show that when there are no repetitions, i.e., when the pattern is $lhlh$, it also has strict width 2. The general case of $i = 2$, i.e., $l^+h^+l^+h^+$, will be considered in section 4 and shown to be always either polynomial or NP-complete. The case of $i = 3$ will be shown to be NP-complete in section 3. This will complete the classification.

For the two cases that we show have strict width 2 by means of an appropriate 2-mapping g, it is again sufficient to define g when all three arguments are in the same level.

THEOREM 4. *Balanced cycles of the type l^+h^+ have strict width* 2.

Proof. Let l_1 and l_2 be the first and last l's in l^+h^+. Then the cycle consists of a path p_1, from l_1 to l_2, that encounters no other l's, but encounters one or more h's, and a path p_2 from l_1 to l_2 that may encounter more l's (although possibly $l_1 = l_2$) but does not encounter any h's. Number the vertices $0, 1, \ldots, k_1$ on p_1 without assigning a number to l_2 if $l_1 = l_2$ and number the vertices $0, 1, \ldots, k_2$ on p_2 without assigning a number to either l_1 or l_2. For the numbered vertices i on p_1, assign them the pair $(i, 0)$. Let j be a specific numbered vertex on p_1 of level h. To the numbered vertices i on p_2, assign the pair (j, i) and consider the lexicographic ordering on pairs. Once again, it can be seen that edges don't cross. Therefore the function $g(x, y, z) = \text{median}(x, y, z)$, where the median is taken in the lexicographic ordering, and x, y, z belong here to the same level, proves the 2-mapping property. □

THEOREM 5. *Balanced cycles of the type lhlh have strict width* 2.

Proof. Assign subindices to the two occurrences of l and of h, so that the type can be denoted by $l_1h_1l_2h_2$. Consider the four paths $1 = l_1h_1$, $2 = l_2h_1$, $3 = l_2h_2$, $4 = l_1h_2$ on the cycle; given three paths $i - 1$, i, $i + 1$ out of these four (modulo 4), the *middle path* is i. The mapping $g(x, y, z)$ with the three arguments at the same level is defined as follows: (0) For levels l and h, it is just majority; (1) If x, y, z belong to three different paths, return the one that belongs to the middle path; (2) If x, y, z belong to the same path, return the one in the middle position (median) on the path; (3) If exactly two out of x, y, z belong to the same path, return the one among the two occurring earliest on the path. This definition satisfies the 2-mapping property. □

3. Some NP-complete cases. We saw in the last section that unbalanced cycles are polynomial and that balanced cycles can be categorized by their type $(l^+h^+)^k$ with $k \geq 1$. We saw that the case $k = 1$ is polynomial; the case $k = 2$ is left for the next section, although we saw that the special case *lhlh is* polynomial. Here we show that the case $k \geq 3$ is NP-complete. The reduction is from k-colorability.

Given an oriented path, we assign levels to the vertices of the path so that if (i, j) is an edge on the path, then $\text{level}(j) = \text{level}(i) + 1$. We say that a path is of *type* r if it starts on level 0, it ends on level r, and all intermediate vertices are on levels i with $0 \leq i \leq r$. The following lemma was also proved in [11].

LEMMA 1. *Let p_1 and p_2 be two paths of type r. Then there is a path p of type r that maps homomorphically to p_1 and to p_2 in such a way that the starting vertex of p maps to the starting vertices of p_1 and p_2, and the ending vertex of p maps to the ending vertices of p_1 and p_2.*

Proof. The proof is first by induction on r and then by induction on the number of vertices of p_1 and p_2 for r fixed. Suppose that neither p_1 nor p_2 have vertices at

level 0 other than their starting vertices. Then we can make p advance on both paths to level 1 and reduce the problem to type $r-1$.

So we may assume that at least one of p_1 and p_2, say p_1, has a vertex at level 0 other than the starting vertex. Let v be the earliest occurring vertex on p_1 at level 0 other than the starting vertex. Let w_i be the earliest occurring vertex on p_i at level r, and let u_i be the vertex preceding w_i at level $r-1$. By inductive hypothesis on r, path p may be started by mapping all the way to the u_i in each p_i and then from u_i to w_i by traversing a single edge.

Now we may extend p mapping from w_1 to v and from w_2 to the starting vertex of p_2. This is by induction on the number of vertices on the two paths with r fixed. Finally, we may extend p mapping from v to the ending vertex of p_1 and from the starting vertex of p_2 to the ending vertex of p_2 again by induction on the number of vertices on the two paths with r fixed. □

We shall refer to a path p, constructed as in the preceding lemma so as to map to any number of given paths, as a *generic path*. Given an oriented cycle of type $(l^+h^+)^k$, we can construct a path that can map precisely to the paths joining any two consecutive l and h. This is done by beginning with an edge from l to $l+1$, then a generic path $l+1$ to $h-1$, then an edge from $h-1$ to h. We can also construct paths that allow us to move between l's in the same l^+. This is done with a generic path from l to $h-1$ followed by a generic path from $h-1$ to l. This allows us to think of the oriented cycle as a cycle $(0,1,2,3,\ldots,2k-1)$ with edges oriented from the even number (corresponding to a single l^+) to the adjacent odd numbers (corresponding to single h^+) modulo $2k$.

A digraph H is a *core* unless there is a proper subdigraph H' and a homomorphism from H to H'; if H is not a core, then such an H' which is a core is called the core of H; otherwise, H *is* its own core. Clearly, if H has core H', then digraph homomorphism for H is equivalent to digraph homomorphism for H'. In the case of an oriented cycle H, if it is not a core, then its core is an oriented path and digraph homomorphism is polynomially solvable, as we saw in the previous section.

Therefore we assume that H is a core. Now, if the instance G contains a copy of H, then this copy must map to H by the identity mapping up to isomorphisms of H; since we can disregard isomorphisms in the solution, we can assume that a specific copy of H in G maps to H via the identity mapping and identify this copy with H itself.

We say that a binary relation can be represented in an instance of the problem if there is an instance G of the problem such that for two specific vertices x, y in G, the possible images that these two vertices can simultaneously have in a homomorphism to H are precisely those in the binary relation.

LEMMA 2. *The binary relation relating opposite numbers i and $i+k$ modulo $2k$ with i even on the cycle can be represented in an instance.*

Proof. Join the fixed cycle $(0,1,\ldots,2k-1)$ to a cycle $(0',1',\ldots,(2k-1)')$ by edges joining i to i'. Then either all i' map to $i-1$ or all map to $i+1$. Repeating this construction, we can ensure that either all i'' map to $i'-1$ or all map to $i'+1$; hence, either they all map to $i-2$, all to i, or all to $i+2$. After repeating this construction $k-1$ times, we obtain a cycle that can be in any of the k possible positions of the same parity, and we can then select two opposite vertices on this cycle. □

LEMMA 3. *The binary relation relating any two different i, j between 0 and $2k-1$ with both i, j can even be represented in an instance.*

Proof. Find for i the opposite vertex on the cycle as in the preceding lemma and

then attach to this opposite vertex a path of length $k-2$ ending in j with its edges oriented appropriately. Then j can go to vertices which are also even as i but not to i itself. □

The relation in the lemma relates two out of k colors precisely when they are different; i.e., it is the k-colorability relation. This gives the following theorem.

THEOREM 6. *Balanced cycles of type* $(l^+h^+)^{\geq 3}$ *are NP-complete for cores (and polynomial for noncores).*

4. The remaining cases. The remaining case consists of balanced cycles of the form $l^+h^+l^+h^+$, which we denote, with the purpose of distinguishing groups of l and h, by $l_0^+h_0^+l_1^+h_1^+$. We have seen that the case $l_0h_0l_1h_1$, with no repetitions, is polynomially solvable. We might thus hope to complete the classification by a case analysis on the number of repetitions. One can observe, however, that for each remaining choice of the number of repetitions, there are both polynomially solvable and NP-complete cases. That is, the complexity of the problem depends on the internal structure of the paths themselves joining l and h vertices to each other and among themselves. It turns out that what really matters is whether there exist digraphs that can map to some of these paths but not to certain others. The possible patterns of which subsets of paths can thus be obtained give rise to a bipartite boolean satisfiability instance, and we use here a classification result for such problems, given in the next section, to complete the classification of oriented cycles.

To understand the characterization given below, consider an instance consisting of a digraph G. We can assume that every oriented cycle in G is balanced, since the oriented cycle H to which G must be mapped is balanced. This means that G has a level structure; i.e., we can assign levels to the vertices of G so that if (u,v) is an edge of G, then $\text{level}(v) = \text{level}(u)+1$. Furthermore, the number of levels of G must be no larger than the number of level of the balanced cycle H. Furthermore, if the number of levels of G is smaller, then G does not map to both l and h vertices in H, so G maps to paths in H, and we have seen that this case is polynomially solvable.

We therefore assume that G has the same number of levels as H and label these levels the same as in H. Now consider all the l vertices and all the h vertices in G. Let S be the set of those l vertices that are joined in G to h vertices by an oriented path not going through any other l vertices and those h vertices that are joined in G to l vertices by an oriented path not going through any other h vertices.

If we remove from G the set S, the digraph G becomes disconnected into connected components. Let the *boundary of* a connected component be set of l and h vertices in S incident on the component; i.e., they connect the component to the rest of G. Notice that each vertex in S must be in the boundary of at least one component containing both l's and h's by the way S was defined.

The next step is to ensure that the boundary of each component contains at most two vertices. The easiest case is that of components that have both l and h boundary vertices. Such components must map to one of the four paths in H joining consecutive l and h vertices. These four paths contain only one l and one h vertex, so all l vertices in the component must map to the same l, and similarly for the h vertices. We may thus collapse together the l vertices in the component and similarly for the h vertices. We thus have precisely two boundary vertices.

The more difficult case is that of components that have only h boundary vertices or only l boundary vertices, say, only h boundary vertices. Such components must map to one of the two maximal paths containing h vertices but no l vertex. We know from section 2 that these paths have strict width 2, and that this property is

equivalent to the 2-Helly property. That is, we can replace a component having some number $r > 2$ of boundary vertices, with several copies of the component, one for each choice of two out of the r boundary vertices, where only these two boundary vertices are attached at their proper places in G. The condition on possible images for the boundary vertices as imposed by the single component is the same as the condition imposed by its copies on pairs of boundary vertices. We thus have copies with at most two boundary vertices.

Now if a component contains an l and an h vertex, they must map to an l_x and an h_y vertex, respectively, where x, y are boolean variables, and they range over $\{0, 1\}$. We thus associate with the component the pair of variables (x, y). Note that a choice of values x, y tells us to which path the component must map. The paths out of the four possible choices where such a component can map gives us a boolean constraint on x, y.

Now consider the case of a component whose at most two boundary vertices are h vertices; the case of l vertices is similar. Say there are two boundary h vertices; the case of a single boundary h vertex is a special case. This component must map to one of two possible maximal paths without l vertices. The two boundary vertices correspond to two pairs (x, y) and (x', y') of boolean variables as described above, since they are also boundary vertices of components having both l and h vertices. We must have $y = y'$. We thus get a boolean constraint on x, x', y.

We have assumed that the x, y come in pairs. However, it is easy to state $y = y'$ for two pairs, (x, y) and (x', y'), by requiring vertices to be h vertices from the same h^+. This is done as in Lemma 1 in the preceding section by connecting two h vertices by a generic path that does not reach level l. We can similarly state $x = x'$. Thus we can assume that variables x and y come separately, in two different sets of boolean variables, and that we have a collection of boolean constraint types of one of the three forms $R(x, y)$, $R(x, x', y)$, and $R(x, y, y')$. Boolean constraint-satisfaction problems were classified as polynomial or NP-complete by Schaefer [10]. This kind of bipartite boolean constraint-satisfaction problem, with two different types of boolean variables, will be classified later as polynomial or NP-complete, as an extension of Schaefer's result. This gives the following theorem.

THEOREM 7. *Balanced cycles of type $l^+h^+l^+h^+$ are either polynomially solvable or NP-complete.*

To decide the complexity of a problem, we must determine which are the boolean constraints of a given type that can be stated by an instance, that is, which relations on two vertices can be stated by an instance of the problem. This question is polynomially decidable, for a chosen such relation, by deciding for each choice of one of r solutions for the two vertices, where all the other vertices will map, in n^r possible ways, where n is the number of vertices in H. We can thus construct an instance G with n^r vertices and all the constraints (edges) consistent with where the vertices map in the r solutions. Then check that these are the only possible solutions for the resulting instance, as far as the two special vertices are concerned. Polynomiality follows from the fact that r is here a constant, the components thus defined map to paths, as described before, and the problem for paths is polynomial.

In Schaefer's classification, there are three polynomially solvable cases, namely, Horn clauses, 2SAT, and linear equations modulo 2. In terms of the approach of Feder and Vardi [1], the first two are explained by Datalog, i.e., of bounded width, while the third is explained by group theory; i.e., it is a subgroup problem. All the previous cases of oriented cycles that were shown to be polynomial belong to the bounded

width case. Is it possible that the polynomial cases in the last theorem include cases that are not of bounded width, yet polynomial because of group theory? For this to be the case, it would have to happen that some relation $R(x, x', y)$ is either $x+x'+y = 0$ or $x + x' + y = 1$ modulo 2. In either case, for a fixed y, we would have a constraint $x + x' = 1$ modulo 2. That is, a component with two h boundary vertices maps the two boundary vertices to two different boundary vertices of the h^+ sector in the two possible ways. However, then the two boundary vertices are separated by vertices that map to the same vertex in both solutions. (Just consider how a path joining the two boundary vertices can map in the two directions and infer that some vertex on the path must map to the same vertex in both solutions.) Then the two solutions can be combined to make the two boundary vertices map to the same vertex, a contradiction. Therefore all polynomial problems are of bounded width.

THEOREM 8. *All oriented cycles define digraph homomorphism problems that are either of bounded width, and hence polynomially solvable, or NP-complete.*

5. The k-partite boolean satisfiability problem. Schaefer showed that all satisfiability problems on a boolean domain are either polynomial or NP-complete [10]. Here we consider a k-partite generalization of the problem. There are k types of variables, each of which ranges over the boolean domain $S_i = \{0_i, 1_i\}$ for $1 \leq i \leq k$. There are also a number of constraint types $R(x_1, x_2, \ldots, x_r)$ that can be imposed on r variables at a time, where x_j ranges over some corresponding S_i.

We show that such k-partite satisfiability problems are also either polynomial or NP-complete. As an application of the bipartite case of this classification, we obtain the above classification of homomorphisms to oriented cycles as polynomial or NP-complete for each choice of an oriented cycle.

We first consider the bipartite case. Suppose that variables $x, x', x'', \ldots$ range over $A = \{0_A, 1_A\}$ and variables $y, y', y'', \ldots$ range over $B = \{0_B, 1_B\}$. If the constraints contain one of the two conditions $x = y$ or $x \neq y$, then we can identify each variable in A with a variable in B, and vice versa, so we actually have a single domain $A = B = \{0, 1\}$ and we can apply Schaefer's classification result to infer that the problem is either in P or NP-complete. Therefore we assume that neither $x = y$ nor $x \neq y$ can be stated.

What then are the binary relations involving one element from A and one from B? They can only be 2SAT clauses. Say the clause $x \vee y$ is present, while the other cases are analogous. Then the clause $\neg x \vee \neg y$ is not present; otherwise, both combined would give $x \neq y$. Similarly, at most one of the two clauses $x \vee \neg y$ and $\neg x \vee y$ can be present; otherwise, both combined would give $x = y$. Say only $x \vee \neg y$ out of these two may be present. Then the at most two clauses present involve the variable x positive (no negation).

We show that at least one 2SAT clause must be present; otherwise, the problems defined by A and B are independent, so the problem is in P if both the A and B parts are in P and NP-complete if at least one of them is NP-complete. For suppose that the two parts are not independent. Then there is a relation R such that $R(x, y')$ and $R(x', y)$ hold, but $R(x, y)$ does not, where the x's and y's are vectors of variables. Replace x and y by minimal subvectors such that $R(x, y)$ still does not hold when the suppressed variables are free. This means that complementing one value in x or y makes R true. Now set all but one of the variables in x to their assigned value in x, leaving the single value x_1, and similarly set all but one of the variables in y to its assigned value in y, leaving the single value y_1. Then $R'(x_1, y_1)$ is false, but both $R'(x_1, \neg y_1)$ and $R'(\neg x_1, y_1)$ are true, giving, say, if $x_1 = y_1 = 0$, either $x \vee y$ or $x \neq y$.

Therefore we assume that $x \vee y$ is present, that $x \vee \neg y$ may be present, and that no other binary condition involving A and B is present. We claim that after suppressing all the variables whose values are forced to 0 or 1, a solution exists iff a solution exists after setting all variables x in A to 1. Therefore the satisfiability problem is equivalent to the satisfiability problem for B, hence either in P or NP-complete, by Schaefer's result.

To prove the claim, suppose to the contrary that there is a solution to the problem that is no longer a solution when all x variables from A are replaced by 1. Consider a condition $R(x_1, \ldots, x_r, y_1, \ldots, y_s)$ such that when we assign 1 to the x_i and the values v_i from the solution to the y_i, we end up having $R(1, \ldots, 1, v_1, \ldots, v_s)$ false. Replace a maximal number of the $x_i = 1$ by free variables, so that the resulting relation $R(1, \ldots, 1, x_{t+1}, \ldots, x_r, v_1, \ldots, v_s)$ is still false for all values of x_i. Simplifying the notation by suppressing the free variables x_i, we have $R(1, \ldots, 1, v_1, \ldots, v_s)$ false, yet R becomes true when one of the first t 1's is replaced by 0. If $t = 0$, then $R(v_1, \ldots, v_s)$ false indicates that these are forbidden values in B, contrary to the assumption that there is a solution with those values. If $t = 1$ and $s = 0$, then $R(1)$ is false, so the single argument of R was forced to 0 in the preprocessing stage. Suppose $t = 1$ and $s \geq 1$. We thus have $R(1, v_1, \ldots, v_s)$ false but $R(0, v_1, \ldots, v_s)$ true. Again replace a maximal number of v_i by free variables y_i, so that $R(1, v_1, \ldots, v_u, y_{u+1}, \ldots, y_s)$ is still false for all values of y_i. Simplifying the notation by suppressing the free variables y_i, we have that $R(1, v_1, \ldots, v_u)$ is false but becomes true when any of the $u + 1$ arguments is complemented. Fix the last $u - 1$ at their assigned v_i. We then have $R(1, v_1)$ false but $R(1, \neg v_1)$, $R(0, v_1)$ true, so if, say, $v_1 = 0$ we obtain either $\neg x \vee y$ or $x = y$, two cases that were not allowed. The remaining case has $t \geq 2$. Fix $t - 2$ of the 1's to 1, and fix all remaining variables to the given v_i, so that we get $R(1, 1)$ false and $R(0, 1)$, $R(1, 0)$ true. This gives either $\neg x \vee \neg x'$ or $x \neq x'$, which combined with the clause $x' \vee y$ gives $\neg x \vee y$, a type of clause that was not allowed. This completes the proof.

THEOREM 9. *All cases of bipartite satisfiability are either in P or NP-complete.*

To generalize the argument to the k-partite case, where there are k boolean sets S_i, introduce two literals x_i and $\neg x_i$ for each S_i; write each 2SAT clause involving two sets S_i and S_j as two implications, say, $x_i \vee x_j$ is written as $\neg x_i \rightarrow x_j$ and $\neg x_j \rightarrow x_i$; and view the resulting implications as a digraph. If the digraph contains a cycle, then equality (or complementarity) can be forced among variables in the cycle, thus reducing the number of parts by collapsing together the S_i in the cycle. If the digraph is acyclic, then some literal, say $\neg x_1$, is a source, in which case x_1 is a sink. Thus all clauses involving x_1 are of the form $x_1 \vee x_i$ or $x_1 \vee \neg x_i$. Again we can set all variables in S_1 to 1 and reduce the problem to a problem with $k - 1$ parts. If S_1 involves no clauses, then it is an isolated part, and its constraints are separate from the rest of the problem. Thus the problem reduces to the union of disjoint problems, it is polynomial if all resulting parts are polynomial and NP-complete otherwise if some part is NP-complete.

THEOREM 10. *All cases of k-partite satisfiability are either in P or NP-complete.*

Acknowledgments. The author had valuable conversations with Yossi Azar and Carlos Subi.

REFERENCES

[1] T. FEDER AND M. Y. VARDI, *The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory*, SIAM J. Comput., 28

(1998), pp. 57–104.
[2] W. Gutjahr, *Graph Colourings*, Ph.D. Thesis, Free University, Berlin, 1991.
[3] W. Gutjahr, E. Welzl, and G. Woeginger, *Polynomial graph colourings*, Discrete Appl. Math., 35 (1992), pp. 29–46.
[4] P. Hell and J. Nešetřil, *On the complexity of H-coloring*, J. Combin. Theory Ser. B, 48 (1990), pp. 92–110.
[5] P. Hell, J. Nešetřil, and X. Zhu, *Duality and polynomial testing of tree homomorphisms*, Trans. Amer. Math. Soc., 348 (1996), pp. 1281–1297.
[6] P. Hell, J. Nešetřil, and X. Zhu, *Complexity of tree homomorphisms*, Discrete Appl. Math., 70 (1996), pp. 23–36.
[7] P. Hell, J. Nešetřil, and X. Zhu, *Duality of graph homomorphisms*, in Combinatorics, Paul Erdös is Eighty, Bolyai Soc. Math. Stud. 2, János Bolyai Math. Soc., Budapest, 1996, pp. 271–282.
[8] P. Hell and X. Zhu, *Homomorphisms to oriented paths*, Discrete Math., 132 (1994), pp. 107–114.
[9] P. Hell and X. Zhu, *The existence of homomorphisms to oriented cycles*, SIAM J. Discrete Math., 8 (1995), pp. 208–222.
[10] T. J. Schaefer, *The complexity of satisfiability problems*, in Proceedings of the 10th ACM Symposium on Theory of Computing, 1978, pp. 216–226.
[11] X. Zhu, *A simple proof of the multiplicativity of directed cycles of prime power length*, Discrete Appl. Math., 36 (1992), pp. 313–315.
[12] X. Zhu, *A polynomial algorithm for homomorphisms to oriented cycles*, J. Algorithms, 19 (1995), pp. 333–345.