Width One, Bounded Strict Width, and Chordal Extensible Graphs

Tomás Feder
268 Waverley St., Palo Alto, CA 94301, USA
tomas@theory.stanford.edu
and
Pavol Hell
School of Computing Science
Simon Fraser University
Burnaby, B.C., Canada V5A 1S6
pavol@cs.sfu.ca

1 Introduction

2 Preliminaries

We start with some definitions regarding constraint satisfaction as studied by Feder and Vardi [5] and others. A *vocabulary* is a set $V = \{(R_1, k_1), \ldots, (R_t, k_t)\}$ of relation names and their arities. A *relational structure* over the vocabulary V is a set S together with relations R_i of arity k_i on the set S (that is $R_i \subseteq S^{k_i}$). An instance of constraint satisfaction is given by a pair I (the *instance*) and T (the *template*) of finite relational structures over the same vocabulary. 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. In what follows, we shall use the homomorphism and variable-value views interchangeably in defining constraint satisfaction problems.

Feder and Vardi [5] defined CSP T as the constraint satisfaction problem where the instance I is given as the input and the template T is fixed. The main question is to determine for which templates T the problem CSP T is polynomial time solvable, and for which templates T the problem CSP T is NP-complete. A large class of polynomial time solvable problems CSP T is given by the problems whose complement has a Datalog program. Problems having such a Datalog characterization are said to have *bounded width*.

Two subclasses of the bounded width case whose recognition is known to be decidable are the problems of *width one*, where the corresponding Datalog program has rules with monadic relations (of arity 1) in the head, and the problems of *bounded strict width*, where a problem has strict width k if the corresponding Datalog program has rules with relations of arity k in the head and after the Datalog program guarantees that there is a solution a greedy assignment to the variables in any order finds a solution. We shall give two alternative characterizations for width one and bounded strict width from [5].

Let S be a connected structure. An element x is an *articulation element* if the structure S can be decomposed into two nonempty substructures that share only x. If we decompose a structure into substructures by identifying all its articulation elements, we say that the resulting substructures without articulation elements are *biconnected components*. A *tree* is a structure whose biconnected components consist each of a single relation occurrence. Following and generalizing the terminology of Hell, Nešetřil and Zhu [6], we say that CSP T has *tree duality* if: A structure S can be mapped to T if and only if every tree that can be mapped to S can be mapped to T.

Given a problem T, let U be the structure defined as follows. The elements of U are the nonempty subsets A of the elements of T. For a relation R of arity k in T, include $(A_1, A_2, \ldots, A_k)$ in R for U, where the A_i are not necessarily distinct, if for every $1 \leq i \leq k$ and every a_i in A_i there exist elements a_j in the remaining A_j such that $(a_1, a_2, \ldots, a_k) \in R$ in T. A *set operation* for CSP T is a homomorphism f from U to T.

Theorem 1 *[5] The following three conditions are equivalent:*
(1) CSP T has width one;
(2) CSP T has tree duality;
(3) CSP T has a set operation.

Strict width k for a CSP T problem can also be captured in two other ways. First, we can require that if we have an instance I and after assigning specific values from T to some of the variables we obtain an instance with no solution, then some k out of the specific value assignments chosen are sufficient to give an instance with no solution. We refer to this property as the *k-Helly property*. Second, we could require that there exists a function g that maps $k+1$ elements from the domain of the structure T to another element, with the property that if all but at most one of the $k+1$ arguments are equal to some value b then the value of g is also b, and, furthermore, for all relations R_i in T, if we have $k+1$ tuples in R_i, then the tuple obtained by applying g component-wise also satisfies R_i. We call such a mapping a *near-unanimity operation* of arity $k+1$. In the special case $k=2$ we call such a mapping a *majority operation.*

Theorem 2 *[5] The following three conditions are equivalent:*
(1) CSP T has strict width k;
(2) CSP T has the k-Helly property;
(3) CSP T has a near-unanimity operation of arity $k+1$.

Given a CSP T problem, we define four problems. The problem HOM T is the same as CSP T. The retract problem RET T has instances I that are instances of CSP T where some of the elements of I may be preassigned values in T. The list homomorphism problem LIST-HOM T has instances I that are instances of CSP T where some of the elements of I may be restricted to arbitrary subsets of the elements of T. The one-or-same list homomorphism problem OS-LIST T has instances I that are instances of CSP T where for any subset S of the elements of T, all elements of I are restricted either to S or to an element of S as possible values.

Given a RET T problem, a *minimal obstruction* is an instance I that does not map to T and such that every substructure J of I maps to T.

Larose et al. [7] define a subcase of near-unanimity operations known as *special near-unanimity operations.*

Theorem 3 *[7] The following three conditions are equivalent:*
(1) RET T has a special near-unanimity operation;
(2) RET T is first-order definable;
(3) RET T has finitely many minimal obstructions.

Theorem 4 *[11] If any of the three equivalent conditions for RET T from Theorem 3 holds, then RET T has width one.*

Theorem 5 *[10] Let H be a reflexive graph. If H has a near-unanimity operation then H has a special near-unanimity operation, has width one, and is dismantlable.*

For partial orders P, the problem RET P shall be restricted to instances that are partial orders, thus transitively closed. The following does not hold if this restriction is not assumed, as in that case arbitrarily large minimal obstructions can be obtained.

Theorem 6 *[8] The following two conditions are equivalent for partial orders P:*
(1) RET P has a near-unanimity operation;
(2) RET P has finitely many obstructions.

Theorem 7 *[9] If any of the two equivalent conditions from Theorem 6 for a partial order P holds, then RET P has width one and P is dismantlable.*

3 Bipartite Graphs

Encode RET H for a bipartite graph $H = (U, V, E)$ with edges oriented from U to V as RET P for a partial order P. The elements of P each have corresponding bipartite cliques $A \times B$ in H. The elements $u \in U$ have clique $\{u\} \times N(u)$, where $N(u)$ denotes the neighbors of u, and these elements are minimum elements in P; the elements $v \in V$ have clique $N(v) \times \{v\}$, where $N(v)$ denotes the neighbors of v, and these elements are maximum elements in P. The remaining bipartite cliques $A \times B$ have $|A|, |B| \geq 2$ and are the maximal such bipartite cliques. Let $A \times B \leq A' \times B'$ if $A \subseteq A'$ and $B' \subseteq B$.

Theorem 8 *If RET H for a bipartite graph H with edges oriented from one side to the other is encoded as RET P for a partial order P, then RET H and RET P are polynomial time reducible to each other.*

Proof. An instance $G = (C, D, F)$ of RET H can be viewed as a partial order with $x \leq y$ if x and y are adjacent vertices in C and D respectively, with the mapped elements in C and D being mapped to the corresponding elements in P. It can be assumed that elemens in C and D will map to minimum and maximum elements corresponding to U and V respectively, establishing the correspondence.

Conversely, given an instance Q of RET P, for an element x of Q assigned to $A \times B$ with $|A|, |B| \geq 2$, we can instead require $y \leq x$ for the corresponding $|A|$ elements y in P associated with elements from U, and require $x \leq z$ for the corresponding $|B|$ elements z in P associated with elements from V. It then suffices to map the minimum and maximum elements of Q to minimum and maximum elements in P. The remaining elements will then immediately map to bipartite cliques $A \times B$, which can then in turn be mapped to corresponding maximal bipartite cliques. To map the minimum and maximum elements, we may consider the problem as an instance G of RET H. $\square$

Theorem 9 *If RET H for a bipartite graph H with edges oriented from one side to the other is encoded as RET P for a partial order P, then RET H has bounded strict width if and only if RET P has bounded strict width.*

PROOF. From the construction in Theorem 8, if an instance G of RET H without solution has a solution for all choices of at most k of its assigned vertices, then the corresponding bipartite partial order Q for RET P from Theorem 8 has no solution, yet has a solution for all choices of at most k of its assigned vertices. Thus if P has strict width k (satisfies the k-Helly property) then H has strict width k (satisfies the k-Helly property).

Conversely, if an instance Q of RET P without solution has a solution for all choices of at most k of its assigned vertices, then the corresponding bipartite instance G of RET H from Theorem 8 has no solution, yet has a solution for all choices of at most k of its assigned vertices. Thus if H has strict width k (satisfies the k-Helly property) then P has strict width k (satisfies the k-Helly property). □

Theorem 10 *If RET H for a bipartite graph H with edges oriented from one side to the other is encoded as RET P for a partial order P, then RET H has finitely many minimal obstructions if and only if RET P has finitely many minimal obstructions.*

PROOF. Bounding the size of the minimal obstructions for RET H bounds the size of the minimal obstructions for RET P and viceversa, by using the correspondence between instances in both directions from Theorem 8. □

Theorem 11 *If RET H for a bipartite graph H with edges oriented from one side to the other is encoded as RET P for a partial order P, then RET H has width one if and only if RET P has width one.*

PROOF. If an instance G of RET H without a solution corresponds to an instance Q of RET P without a solution, and a tree mapping to Q but not to P is obtained by tree duality, we infer a tree mapping to G but not to H. So if P has tree duality then H has tree duality.

Conversely, if an instance Q of RET P without a solution corresponds to an instance G of RET H without a solution, and a tree mapping to G but not to H is obtained by tree duality, we infer a tree mapping to Q but not to P. So if H has tree duality then P has tree duality. □

From Theorems 3, 6, 9, and 10 we infer the following.

Corollary 1 *Let H be a bipartite graph with edges oriented from one side to the other. If RET H has a near-unanimity operation then RET H has a special near-unanimity operation.*

From Theorems 7, 9, and 11 we infer the following.

Corollary 2 *Let H be a bipartite graph with edges oriented from one side to the other. If RET H has a near-unanimity operation then RET H has width one and is dismantlable.*

Lemma 1 *Consider a graph H. Let H' be a graph having H as an induced subgraph such that for every vertex x in H', there is a vertex y in H that dominates x (that is $N_{H'}(x) \subseteq N_{H'}(y)$). Then, if H has width one for LIST-HOM H, then H' has width one for RET H'; if H' is bipartite with edges oriented in one direction and H has width one for LIST-HOM H, then H' has width one for RET H'.*

PROOF. Given an instance G' for RET H', carry out the arc consistency procedure on G'. If an empty list is obtained, then there is no solution. Otherwise we may remove from G' any vertex u having list $\{x\}$ of size one, since the neighbors of u will have list contained in the neighbors of x in H'. For the remaining vertices u, if their list constains a vertex x not in H, then the list also contains a vertex y in H that dominates x, so we may remove x from the list without affecting whether the problem has a solution. The resulting lists will be nonempty, contained in H, and also satisfy the arc consistency condition, so since LIST-HOM H has width one, the problem has a solution. Therefore the arc consistency procedure for correctly decides whether G' has a solution, and so RET H' has width one.

In the bipartite case, for $H' = (U, V, E)$, the instance G' for RET H' is bipartite, with the vertices in the two sides of G' having lists equal to all of U and V respectively, or to single vertices. As before, we carry out the arc consistency procedure; if the resulting lists are nonempty, remove vertices from G' with lists of size one, and remove vertices x not in H from the remaining lists because they also contain a vertex y in H dominating x. Since LIST-HOM H has width one, it follows that this instance has a solution. Therefore the arc consistency procedure correctly decides whether G' has a solution, and so RET H' has width one. □

Proposition 1 *Let H be given by $V(H) = \{a, b, c, d\}$ and $E(H)$ consisting of all possible edges and loops except for (a, d). Obtain H' by adding to H vertices u and v with loops and with edges joining u to all vertices of H except for c, and edges joining v to all vertices of H except for b. The graph H' is dismantlable. Then H' has width one for RET H', but does not have strict width k for any k. Let H'' be the bipartite graph having two vertices x_1, x_2 for every vertex x in H', and edges x_1y_2 and y_1x_2 for every edge x, y in H'. Then H'' has width one for RET H'', but does not have strict width k for any k.*

PROOF. H has width one for LIST-HOM H. Then by the Lemma, H' has width one for RET H'. The graph H' has strict width k if and only if it has a near-unanimity function g with $k + 1$ arguments. Let x^k denote a sequence of k x's.

Note first that $g(a^lbd^{k-l}) \in \{a, d\}$ for all $0 \le l \le k$, since it is adjacent to $g(u^lbu^{k-l}) = u$ and to $g(v^lbv^{k-l}) = v$.

Note next that $g(a^lbd^{k-l}) = g(a^{l+1}bd^{k-(l+1)})$, since these two values of g must be adjacent and a is not adjacent to d.

Consequently, $g(a^lbd^{k-l}) = g(a^{l'}bd^{k-l'})$ for all $0 \le l, l' \le k$. Setting $l = 0$ and $l' = k$, we get $d = g(bd^k) = g(a^kb) = a$, a contradiction.

Consider now the bipartite graph $H'' = (U, V, E)$. Consider an instance G'' of RET H'' that is bipartite, with the vertices in the two sides of G'' having lists equal to all of U and V respectively, or to single vertices. Such an instance G'' corresponds directly to an instance G' of RET H' by identifying two vertices x_1, x_2 in H'' with the corresponding vertex x in H'. Since H' has width one for RET H', the arc consistency procedure correctly decides whether the instance has a solution, so H'' has width one for RET H''.

To show that H'' does not have strict width k for any k, show as before that $g(a_i^lb_id_i^{k-l}) \in \{a_i, d_i\}$ for all $0 \le l \le k$ and $i = 1, 2$, and equals a_i for $i = 1$ if and only if it equals a_i for $i = 2$. Then as before we obtain $d_i = g(b_id_i^k) = g(a_i^kb_i) = a_i$, a contradiction. □

Proposition 2 *There exist dismantlable bipartite graphs H_1, H_2 with edges oriented from one side to the other such that RET H_1 is NP-complete and RET H_2 has width one but does not have bounded strict width.*

PROOF. We only give a sketch of the proof. A CSP T problem is encoded in [5] as a polynomially equivalent problem RET H for a dismantlable graph H. The encoding of an NP-complete CSP T gives an NP-complete RET H with H dismantlable. The encoding preserves tree duality and thus width one.

The proof that H does not have strict width k for any k uses the fact that a series of assigned vertices in H corresponds to a series of equality constraints on elements of the instance for T, and if this series joins two different elements, then the arbitrarily many equality constraints on elements of the instance for T in the series are needed to show that there is no solution for the instance, corresponding to arbitrarily many assigned vertices in H, thus more than k. □

It was shown in [3] that LIST H for bipartite graphs is either polynomial with majority operation or NP-complete.

Theorem 12 *If H is a bipartite graph with edges oriented from one side to the other such that LIST H has a majority operation then LIST H has width one.*

PROOF. The bipartite graphs that have a majority operation are the complements of circular arc graphs of clique cover number two. The vertices of $H = (U, V, E)$ correspond to intervals in a circle containing two distinct points N and S that split the circle into a left side and a right side. The intervals for elements of U contain N but not S, and the intervals for elements of V contain S but not N. A vertex u in U is adjacent to a vertex v in V if the corresponding intervals do not intersect.

We define a set operation on the sets contained in U and those contained in V. For subsets of U, select the element u whose corresponding interval has the endpoint in the left side closest to N; For subsets of V, select the element v whose corresponding interval has the endpoint in the right side closest to S.

Suppose x with list contained in U, and y with list contained in V, are adjacent in G, and are so mapped to u and v respectively. If u and v are not adjacent in H, then intervals for u and v intersect. Say they intersect in the left side. Then, by the choice of u, the interval for v intersects in the left side all intervals for elements of the list for x, so v is not adjacent to any element of the list for x and would have been removed from the list for y by the arc consistency procedure. The case where the two intervals intersect in the right side leads analogously to a contradiction, exchanging the roles of x and y, of u and v, of N and S, and of left and right. This completes the proof. □

4 Reflexive Graphs

Encode RET R for a reflexive graph R with edges oriented from U to V as RET H for a bipartite graph $H = (A, B, E)$. Here $A = V(R)$, and B consists of the maximal cliques of R, with E joining vertices in A to the maximal cliques in B that they belong to.

Theorem 13 *If RET R for a reflexive graph R is encoded as RET H for a bipartite graph H with edges oriented from one side to the other, then RET R and RET P are polynomial time reducible to each other.*

PROOF. An instance G' of RET H' corresponds to an instance G of RET H having a vertex x that must map to A for each vertex x' in G', and a vertex z adjacent to both x and y corresponding to adjacent vertices x' and y' in G'. Clearly z must map to a vertex in B corresponding to a maximal clique in G', so the vertices x' and y' are mapped to adjacent vertices in G' by this reduction.

Conversely, given a bipartite instance G of RET H, the vertices in G that must map to A correspond to vertices in an instance G', and the vertices in G that must map to B define a clique in G'. If a vertex in G is preassigned a specific vertex u in B, then all its neighbors are required to be adjacent to all the vertices of the maximal clique corresponding to u, and will thus map to some vertex in that maximal clique. When G' is mapped to H', the vertices z of G that define a clique in G' will get a clique in H', and can therefore be mapped to the vertex corresponding to a maximal clique containing this clique. □

The proofs of the next three theorems are derived analogously to Theorems 9, 10, and 11.

Theorem 14 *If RET R for a reflexive graph R is encoded as RET H for a bipartite graph H with edges oriented from one side to the other, then RET R has bounded strict width if and only if RET H has bounded strict width.*

Theorem 15 *If RET R for a reflexive graph R is encoded as RET H for a bipartite graph H with edges oriented from one side to the other, then RET R has width one if and only if RET H has width one.*

Theorem 16 *If RET R for a reflexive graph R is encoded as RET H for a bipartite graph H with edges oriented from one side to the other, then RET R has finitely many minimal obstructions if and only if RET H has finitely many minimal obstructions.*

As in the previous section we can derive Theorem 5 as well from this construction.

Proposition 3 *There exist dismantlable reflexive graphs R_1, R_2 such that RET R_1 is NP-complete and RET R_2 has width one but does not have bounded strict width.*

PROOF. Let H be a dismantlable bipartite graph as in Proposition 2. The vertices of R consist of U, V, and three additional vertices a, b, c; the edges of R consist of all loops, all edges of E, and the edges of four cliques on $U \cup \{a\}$, $V \cup \{b\}$, $U \cup \{c\}$, and $V \cup \{c\}$. Thus elements x, y in an instance for R with edges ax, xy, yb require x, y to map to adjacent elements from U and V; we can assume that in an instance for R, the elements adjacent to a map to U, the elements adjacent to b map to V, and all remaining elements map to c. We can dismantle R by first removing a and b which are dominated by their neighbors in U and V respectively, then remove all vertices other than c which are now dominated by c.

The claims about NP-completeness, width one, and bounded strict width carry over from Proposition 2. □

It was shown in [2] that LIST R for reflexive graphs is either polynomial with majority operation or NP-complete.

Theorem 17 *If R is a reflexive graph such that LIST R has a majority operation then LIST H has width one.*

PROOF. The reflexive graphs that have a majority function are the interval graphs. The vertices of H correspond to intervals in the real line, with two vertices adjacent if the corresponding intervals intersect. We define a set function. Otherwise, for each vertex Select from each nonempty subset the element u.

Suppose x and y are adjacent in G and are so mapped to u and v respectively. If u and v are not adjacent in H, then intervals for u and v do not intersect, say the interval for u is to the left of the interval for v. Then, by the choice of v, the interval for u is to the left of all intervals for elements of the list for y, so u is not adjacent to any element of the list for y and would have been removed from the list for x by the arc consistency procedure. This contradiction completes the proof. □

It was shown in [4] that LIST H for graphs possibly having both loops and non-loops is either polynomial with majority operation or NP-complete. We do not know if these problems have width one as well.

5 Chordal-Extensible Graphs

The following result follows essentially from the work in [2, 1]. The CONNECTED-LIST H problem is the LIST H problem where the lists are restricted to subsets of vertices that induce connected subgraphs of H. This is strictly speaking not a CSP problem, as the intersection of connected lists is not necessarily a connected list. However, it can be viewed as a CSP problem by making copies of the elements so that all the allowed lists are either disjoint or subsets of one another

Theorem 18 *Let R be a reflexive chordal graph. Then CONNECTED-LIST R has width one and strict width k for some $k \leq n = |V(R)|$.*

PROOF. We first describe an algorithm for CONNECTED-LIST H for chordal graphs H, due to Feder and Hell [2]. Let $v_1, v_2, \ldots, v_n$ be a perfect elimination ordering for the vertices of H, so that the neighbors v_j with $j > i$ for each v_i form a clique. Let G be an instance of CONNECTED LIST-HOM H. Map all vertices x in G with list $\{v_1\}$ to v_1. One can then assume that no other vertex y in G maps to v_1, because if its list constains v_1 then it also contains a neighbor of v_1 that dominates v_1, by connectivity. Restrict the lists of all neighbors y of vertices x mapped to v_1, so that the list of y is contained in the neighborhood of v_1, which forms a clique. Remove then all x mapped to v_1 from the instance G, and remove v_1 from H and from the the lists of all vertices z remaining in G. Proceed then with v_2 as before with v_1, and so on. If at any point in time a list becomes empty, then there is no solution. Otherwise, when v_n is reached, a solution is obtained.

Suppose there is no solution for G. We construct a tree T with lists on its vertices that does not map to H but maps to G, so that the list of each vertex t in T is the same as the list of its image x in G. Thus T witnesses the fact that G does not map to H. The existence of such a tree T shows that CONNECTED LIST-HOM H has width one.

The root t of T corresponds to the first vertex x in G whose list becomes empty with the above algorithm; the list of t is the same as the list of x. The children t' of t in T correspond to a set of neighbors x' of x in G that get mapped to corresponding vertices v' in H, causing the algorithm to remove from the list for x some vertices in H occurring in the perfect elimination ordering after the last v_i assigned by the algorithm; the list for each such t' is the same as the list for x'. We proceed to define children t'' of t' in T as before with the children t' of t in T, with corresponding neighbors x'' of x' in G, mapped to v'' and v' respectively, where v'' preceeds v' in the perfect elimination ordering and x'' causing the algorithm to remove from the list for x' some vertices in H occurring in the perfect elimination ordering after v', and so on. If t'' is a child of t' in T, with corresponding vertices x'' and x' in G, that are mapped by the algorithm to v'' and v' respectively, then v'' preceeds v' in the perfect elimination ordering, thus bounding the height of the tree T.

We claim that if a vertex t' other than the root in T corresponds to x' in G mapped to v' in H by the algorithm, then we can assume that different such t' have distinct corresponding v'. Thus the size of T is at most the size of H, and if some vertices x in G were preassigned values in H, then the number of such preassigned values causing G to have no solution is $s \leq n = |V(H)|$, showing that H has strict width equal to the largest such s over instances G.

It remains to establish the claim. We first show that if t'_1 and t'_2 are distinct children of a vertex t in T, then the corresponding assigned v'_1 and v'_2 in H are not adjecent. Otherwise, if say v'_1 preceeds v'_2 in the perfect elimination ordering, or v'_1 equals v'_2, then we only need to include t'_1, and not t'_2, in the tree T, because the corresponding values after v'_2 in the perfect elimination ordering that were removed from the list of x corresponding to t because of v'_2 are also removed from the list of x because of v'_1 by the algorithm.

The same argument shows that for v'_1 and v'_2 in H corresponding to t'_1 and t'_2 as just described, there is no path $v'_1 = u_1, u_2, \ldots, u_m = v'_2$ in H where each u_i preceeds u_{i+1} in the prefect elimination ordering. Indeed, the set of neighbors of u_i after v_2' is contained in the set of neighbors of u_{i+1} after v_2'.

We finally show that there is no vertex u in the perfect elimination ordering with paths $u = u_1, u_2, \ldots, u_{m_1} = v_1'$ and $u = w_1, w_2, \ldots, w_{m_2} = v_2'$ in H where each u_i preceeds u_{i+1} and each w_i preceeds w_{i+1} in the perfect elimination ordering. Otherwise u_2 and w_2 are adjacent in H, say u_2 preceeds w_2 in the perfect elimination ordering, shortening the first path $u' = u_2, \ldots, u_{m_1} = v_1'$ while the second path remains of the same length, $u' = w_1', w_2, \ldots, w_{m_2} = v_2'$. We thus repeatedly shorten one of the two paths, until one of the two paths has length zero, say the first path, at which point the second path goes from v_1' to v_2', which is the case that was analyzed before.

This last result about two paths ending in v_1' and v_1' respectively shows that the subtrees rooted at t_1' and t_2' respectively have corresponding disjoint sets of vertices in H, completing the proof. ∎

It is known that there are reflexive graphs H such that RET H has bounded strict width and H is not in the variety of chordal graphs.

A bipartite graph $H = (U, V, E)$ is *chordal-extensible* if there exists a reflexive chordal graph $K = (W, F)$ with $W = U \cup V$, $E \subseteq F \subseteq E \cup (U \times U) \cup (V \times V)$, where $K_U = (U, F \cap (U \times U))$ and $K_V = (V, F \cap (V \times V))$ are connected subgraphs of K.

Theorem 19 *Let H be a chordal-extensible graph with edges oriented from one side to the other. Then RET H has width one and strict width k for some $k \leq n = |V(H)|$.*

Proof. This follows from Theorem 18 by adding the edges that make H chordal and verifying that the lists corresponding to the two sides U and V of the bipartite graph, are connected. ∎

Corollary 3 *Let C be the variety chordal-extensible graphs. Then all $H \in C$ have extended width one and strict width k for some k, for RET H.*

Conjecture 1 *All graphs H of strict width k for some k for RET H are in the variety of chordal-extensible graphs.*

This conjecture is known for $k = 2$, corresponding to retracts of products of irreflexive paths.

A bipartite graph is *chordal bipartite* if it contains no induced cycle of length at least six. A chordal bipartite graph H with at least one edge has two adjacent vertices x, y such that $N(x) \cup N(y)$ induces a complete bipartite subgraph of H; clearly this property holds for induced subgraphs of H as well. We say that a bipartite graph $H = (U, V, E)$ is a *generalized chordal bipartite graph* if it has an elimination ordering that repeatedly removes either (1) a vertex x in U of degree one, or (2) a vertex y in V that is dominated by some other vertex z in V, until the graph consists of just one vertex.

Proposition 4 *Every chordal bipartite graph is a generalized chordal bipartite graph.*

Proof. Find the two adjacent vertices $x \in U, y \in V$ in the chordal bipartite graph such that $N(x) \cup N(y)$ induces a complete bipartite graph. if $|N(x)| = 1$ we may remove x in step (1); if $|N(x)| \geq 2$ then y is dominated by any vertex in $N(x) - y$ and we may remove y in step (2). ∎

Let $H = (U, V, E)$ be a generalized chordal bipartite graph. Define a reflexive graph $K = (U \cup V, F)$ by adding to H all loops, and (1) all edges u_1u_2 joining pairs u_1, u_2 vertices of U that have a common neighbor v in V; (2) for every vertex v_1 of V, when v_1 is removed in the elimination ordering, there is a vertex v_2 remaining in V that dominates v_1: Add the edge v_1v_2. Thus the subgraph of K induced by V is a tree.

Theorem 20 *The reflexive graph K is chordal, and the subgraphs of K induced by U and by V are connected. Therefore H is a chordal-extensible graph, and RET H, when we also allow lists contained in U or in V that are connected in K, has width one and bounded strict width.*

PROOF. Consider the removal of vertices of H one by one according to the elimination ordering.

(1) When we remove a vertex v_1 in V, the vertex v_1 has only one neighbor v_2 from V in K, where v_2 dominates v_1 in H. All neighbors u of v_1 in U are also adjacent in K. Therefore the neighbors of v_1 in K induce a clique.

(2) Any two vertices u_1, u_2 in U that have a common neighbor in V, will still have a common neighbor in V after a vertex from V is removed from H, because the removed vertex is dominated by a vertex that remains in H. When we remove a vertex u_1 in U of degree one in H, with a single neighbor v in V, all neighbors u of u_1 in U must have a neighbor in common with u_1 in V, so this common neighbor must be v. Therefore all these u are adjacent in K and adjacent to v, and so the neighbors of u_1 in K induce a clique.

Therefore K has a perfect elimination ordering such that each vertex removed has neighbors inducing a clique, that is K is chordal. □

Theorem 21 *If $H = (U, V, E)$ is a generalized chordal bipartite graph, then greedily removing vertices satisfying the conditions for an elimination ordering, choosing among potential vertices to be removed arbitrarily, will indeed produce an elimination ordering. Thus generalized chordal bipartite graphs can be recognized in polynomial time.*

PROOF. For a valid elimination ordering, define the reflexive chordal graph K as before.

If a vertex u in U has degree one, then u can be moved to the beginning of the ordering without affecting the dominations in V required for a valid ordering.

Suppose v_1 is a vertex in V that is dominated by some vertex v_2 that occurs after v_1 in the ordering. Then we can move v_1 to the beginning of the ordering by letting any vertex preceding v_1 that used v_1 as a dominating vertex use v_2 instead.

Alternatively, suppose v_1 in V is dominated by some v_2, but every such v_2 precedes v_1 in the ordering. Consider the last such v_2 in the ordering. When v_2 is about to be removed, all neighbors of v_1 are also neighbors of v_2 and thus are still in the graph, since they have degree at least two. The vertex following v_2 that dominates v_2 must also dominated v_1, so by the choice of v_2 this vertex must be v_1. Therefore, when v_2 is about to be removed, it has the same neighbors as v_1. We may thus exchange v_2 and v_1 in what remains of the ordering and still have a valid ordering. We are then back to the previous case, with v_2 occurring after v_1.

Therefore any valid chice of first vertex can start the elimination ordering, and we proceed inductively on the rest of the ordering, completing the proof. □

Given a reflexive chordal graph K, let the bipartite graph H associated with K be given by letting H have two vertices x_1, x_2 for every vertex x in K, with edges x_1y_2 and y_1x_2 for every edge xy in K.

Theorem 22 *Bipartite graphs H associated with a reflexive chordal graph constitute a family of chordal-extensible graphs incomparable to the generalized chordal bipartite graphs. Such bipartite graphs H can be recognized in polynomial time.*

PROOF. H is chordal-extensible by considering the chordal graph K' obtained from H by adding the edges x_1y_1 and x_2y_2 for every edge xy in K as well. Generalized chordal bipartite graphs are not necessarily of this form since they may have a different number of vertices in the two sides.

Consider the chordal graph K with vertices $a_1, a_2, a_3, b_1, b_2, b_3$, all loops, all edges $a_i a_j$, $a_i b_j$ for $i \neq j$. The bipartite graph H associated with K is not a generalized chordal bipartite graph.

The bipartite graphs H associated with a reflexive chordal graph can be recognized by repeatedly decomposing the set of vertices into connected groups according to degree, until all vertices vertices in a group A have the same degree within A, and the same degree to vertices in another group B. At that point, every group A corresponds to a clique in K, and the union $A \cup B$ of two groups that share edges also corresponds to a clique in K, because every chordal graph that is not a clique has a vertex that strictly dominates another. We may then pair-up vertices in the two sides of each group A arbitrarily to obtain K. $\square$

It remains open whether chordal-extensible graphs can be recognized in polynomial time.

6 Reflexive Chordal Extensible Graphs

If H is a reflexive graph, let $H' = (U, V, E)$ is the bipartite graph obtained from H by letting U be the vertices of H, letting V be the maximal cliques of H, and E joining vertices in U to cliques in H. We have seen that H has bounded strict width if and only if H' has bounded strict width. If H' is chordal extensible, we say that H is reflexive chordal extensible. Thus reflexive chordal extensible graphs have bounded strict width. More generally, if H' is in the variety of chordal extensible graphs, then H is a reflexive graph of bounded strict width.

7 Mixed Graphs Have Connected Loops

It seems harder to relate the properties of bounded strict width and width one for mixed graphs, with some loops and some non-loops. All we know is that the loops must be connected to avoid NP-completeness.

Theorem 23 *Let H be a connected graph with some loops and some non-loops. Suppose the subgraph induced by the vertices that have loops is not connected. Then the RET H is NP-complete.*

PROOF. Let $k \geq 2$ be the number of connected components of the subgraph of H induced by the vertices of H that have loops. Denote these components by $S_0, S_1, \ldots, S_{k-1}$. Denote by T the subgraph induced by the vertices of H that do not have loops.

We can restrict a vertex x in an instance G to map to the vertices in the sets S_i by putting a loop on x. We can restrict two such x, y with loops to map to the same S_i by joining them with a reflexive path of length at least the maximum diameter of components S_i.

We shall gradually reduce the number k of components S_i to which a vertex x with a loop can map, until $k = 2$. For $i \neq j$, let $d(i, j)$ be the minimum distance between a vertex in S_i and a vertex in S_j, and let $d \geq 2$ be the minimum $d(i, j)$ for $i \neq j$.

We can assume all $d(i, j)$ equal d. Otherwise there is a vertex v in some S_i that is at distance d from some vertex in some S_j, but is not at distance d from any vertex in some other S_l. Then an irreflexive path of length d to a vertex x with a loop allows x to map to S_i or to S_j but not to S_l, and by restricting vertices with loops to be in the same S_t as such an x we have reduced k while still satisfying $k \geq 2$.

For each S_i, let K_i be the largest clique in T such that there is a vertex v in S_i to which every vertex in K_i is adjacent, and every vertex w in K_i has an S_j with $j \neq i$ such that w is at distance $d - 1$ from S_j. Let u be a vertex in K_i, and consider a clique containing two vertices x, y in an instance and a clique $(K_i - \{u\}) \cup \{v, x, y\}$. Join x to a vertex z having a loop with an irreflexive path of length $d-1$, and join y to a vertex t having a loop with an irreflexive path of length. Then at most one of z, t can map to an S_j with $j \neq i$, since otherwise the clique $(K_i - \{u\}) \cup \{x, y\}$ would contradict the maximality of K_i. It is possible that both z, t map to S_i, and that say z maps to S_i and t maps to an S_j with $j \neq i$, by mapping x to v, and y to u. In fact we can assume that every S_j with $j \neq i$ can so be reached by z and by t, but not by both simultaneously, since otherwise we can again reduce k by considering the sets S_j that can so be reached.

We can then require, for any two particular S_i and S_j, that at least one of z, t map to S_i, and similarly that at least one of z, t map to S_j. This, in particular, restricts z to map to either S_i or S_j, and to no other S_l. We have thus reduced the number of S_i to which a vertex in an instance can map to $k = 2$. Say the two allowed S_i, are S_0 and S_1.

The above gadget can then be used to enforce that at most one of z, t map to S_1 but not both. If we say that z has value i if it maps to S_i, we have thus obtained a boolean clause $\overline{z} \vee \overline{t}$. Similarly, we can use the gadget to enforce that at most one of z, t map to S_0, we obtain the clause $z \vee t$. Both clauses together allow us to represent any 2-satisfiability clause.

Let r be the size of the clique K_0. Let K by a clique of size $r+2$ with vertices $x_0, x_1, x_2, \ldots, x_{r+1}$. Add a loop on x_0. Join each x_i with $1 \leq i \leq r+1$ to a vertex y_i with a loop by an irreflexive path of length $d-1$. Require all y_i with $2 \leq i \leq r+1$ to map to the same S_i by means of reflexive paths of length the diameter of H. It is then possible for (x_0, y_1, y_2) to have values $(0,0,0)$, $(0,1,0)$, $(0,0,1)$, or $(1,1,1)$, but not values $(0,1,1)$, since in that case the x_i with $1 \leq i \leq r+1$ would form a clique in T of size $r+1$, contrary to the maximality of K_0.

We have thus obtained a condition that does not allow for (x_0, y_1, y_2) to have values $(0,1,1)$, but allows (x_0, y_1) to have values $(0,1)$, or (x_0, y_2) to have values $(0,1)$, or (y_1, y_2) to have values $(1,1)$. That is, this condition cannot be expressed by a 2-satisfiability clause. Combining this condition with all the 2-satisfiability clauses, which were obtained earlier, we obtain a boolean constraint satisfaction problem that is not among the polynomial cases of Schaefer's classification, and is therefore NP-complete. $\square$

8 One-or-same List Problem

For a fixed graph H, the OS-LIST H problem is the list homomorphism problem for H where all instances have a set S of vertices in H such that all lists in the instance are either all of S or a single vertex in S. In other words, the instance must retract to a the subgraph of H induced by the set S.

Theorem 24 *If H is reflexive, then the OS-LIST H problem for H is polynomial if H is chordal and NP-complete otherwise. If H is irreflexive, then the OS-LIST H problem for H is polynomial if H is chordal bipartite and NP-complete otherwise.*

PROOF. The reflexive result was shown by Feder and Hell elsewhere. For the irreflexive case, if H is chordal biparite, then every subgraph of H induced by a subset S is chordal bipartite, and the retract problem for chordal bipartite graphs is polynomial as a special case of generalized chordal bipartite graphs. If H is not chordal bipartite, then some set S of vertices induces a cycle of length $k \neq 4$, and the retract problem for such a cycle is NP-complete. $\square$

We would like to characterize the OS-LIST H list problem when H is a mixed graph, containing some vertices with loops and some vertices without loops. A *mixed chordal* graph is a graph that does not contain (1) an induced path of length at least two with loops at the ends end no loops in between; (2) an induced irreflexive triangle; (3) an induced reflexive four-cycle; (4) an induced cycle of length five or more that may contain loops and non-loops; and (5) an induced subgraph

$K = (V, E)$ with $V = \{a, b, c, d, e, f\}$ and $E = \{aa, bb, ab, ac, ad, be, bf, cd, de, cf, ef\}$. As the five forbidden subfamilies are NP-complete for the retract problem, the following is obtained.

Theorem 25 *If H is not a mixed chordal graph, then OS-LIST H is NP-complete.*

PROOF. We prove case (5). Consider bipartite instances for K, replace a by a_1, a_2, replace b by b_1,b_2, with edges $a_1a_2, a_1b_2, b_1a_2, b_1b_2, a_1c, a_2d, b_1e, b_2f, cd, de, cf, ef$. The resulting graph is a complete bipartite graph with four vertices v_1, v_2, v_3, v_4 and w_1, w_2, w_3, w_4 in each side minus a perfect matching $v_1w_1, v_2w_2, v_3w_3, v_4w_4$. If we restrict the vertices in a bipartite instance to neighbors of v_4 or w_4, we are restricted to the 6-cycle $v_1w_2v_3w_1v_2w_3$, and the retract problem for an irreflexive 6-cycle is NP-complete. □

There is a case similar to (5) that we conjecture is also NP-complete. The graph has vertices a, b, c, d, e, f, g, h and all edges except for the loops at e, f, g, h and the edges af, be, ch, dg, ef, gh.

Let H be a mixed graph, and let $H' = (V_1, V_2, E)$ be the corresponding bipartite graph, with two vertices x_1, x_2 for every vertex x in H, and edges x_1y_2 and y_1x_2 for every edge xy in H. Suppose H' is chordal extensible, that is we can add edges to H' to obtain a chordal graph H'', where the added edges are each contained in V_1 or contained in V_2, and each of the two subgraphs of H'' induced by V_1 and by V_2 is connected. Then the retract problem for H'' is polynomial. Say that H' is *symmetrically chordal extensible* if in addition the chordal graph H'' has an edge x_1y_1 in V_1 if and only if it has the corresponding edge x_2y_2 in V_2.

Theorem 26 *If H' is symmetrically chordal extensible, then RET H is polynomial.*

PROOF. Given an instance G, let G' be the corresponding bipartite graph, where every vertex in G is restricted by V_1, or by V_2, or assigned a single vertex in H'. The algorithm of Feder and Hell for connected list homomorphism for a chordal graph H'' removes vertices from H'' and from lists in G' according to a perfect elimination ordering, mapping vertices from G' to H'' once their lists contains just the vertex being eliminated from H''. By symmetry of H'', we can choose the perfect elimination ordering to be $x_1, x_2, y_1, y_2, z_1, z_2, \ldots$, that is we remove in order a vertex x_1 from V_1 and the corresponding vertex x_2 from V_2. This ensures that when the list for a vertex t_1 in one side of G' contains just x_1, then the corresponding list for t_2 contains just x_2 by symmetry, so we can assign to the vertex t the corresponding vertex x and obtain a solution for G. □

Theorem 27 *If H is a mixed chordal graph, then the subgraph H_R of H induced by vertices in H that have loops is a chordal graph, and the subgraph H_I of H induced by vertices in H that do not have loops is a chordal bipartite graph. Furthermore, if L is a connected component of H_I, and K is the set of vertices in H_R that are adjacent in H to a vertex in L, then K induces a clique in H.*

PROOF. The fact that H_R is chordal and H_I is chordal bipartite follows immediately from the definition of mixed chordal. If K has two vertices x and y that are not adjacent, and p is a shortest path in H from x to y such that all internal vertices of p are in L, and p has loops at x and y, contrary to the definition of mixed chordal graphs. □

Let H be a mixed graph with the structure of Theorem 27. A *fragment* of H is the subgraph induced by $L \cup K$ for a pair L, K as in Theorem 27. The bipartite graph R' associated with a fragment R thus consists of two copies L', L'' of the connected chordal bipartite graph L joined by some edges to the complete bipartite graph K' associated with K. Write $K' = (V_1, V_2, E)$, $L' = (V_1', V_2', E')$, $L'' = (V_1'', V_2'', E'')$, so that R' consists of $K' \cup L' \cup L''$ plus edges joining V_1 to V_2' and V_2'', and edges joining V_2 to V_1' and V_1''. Let the *semi-fragment* R'' be the subgraph of R' induced by the vertices of $K' \cup L'$. We say that the fragment is *special* if R'' is chordal extensible and the corresponding chordal graph R''' with the same vertices as R'' contains all edges joining two vertices in K'.

Theorem 28 *Let H be a mixed graph with the structure of Theorem 27. If every fragment of H is special, then H is symmetrically chordal extensible, and so RETH is polynomial.*

Proof. Let R be a special fragment. Note that the subgraph R'' on $K' \cup L'$ is isomorphic to the subgraph R_1'' on $K' \cup L''$. Combining the chordal graph R''' on $K' \cup L'$ containing all edges joining two vertices in K' with the chorresponding chordal graph R_1''' on $K' \cup L''$ also containing all edges in K' gives a chordal graph T on $K' \cup L' \cup L''$ that makes R' symmetrically chordal extensible. Combining the various chordal graphs T for the various fragments R of H makes the bipartite graph H' associated with H also symmetrically chordal extensible. □

Not all mixed chordal graphs are symmetrically chordal extensible. An example is a fragment whose biparite graph is the complete bipartite graph $K_{2,2}$ and the two vertices on either side of this bipartite graph do not dominate each other. In particular, with this condition on $K_{2,2}$ and a clique with loops on two vertices, we obtain an NP-complete problem. Thus the graph has edges $ac, ad, bc, bd, ee, ff, ef, ae, bf, ce, df$. This encodes the RET H problem on the irreflexive 6-cycle represented as ee, ef, fb, bd, da, ae, with one side of the 6-cycle taken as the neighbors of c and the other side as the neighbors of f, proving NP-completeness for this mixed chordal graph.

Not all fragments R for which the one-or-same list problem is polynomial have an associated bipartite graph R' that is chordal bipartite. An example of this is R consisting of a cycle C of even length $2k$, plus a loop on a vertex x_0 in C, plus edges joining x_0 to all vertices at even distance from x_0 in C. This fragment is special, as can be verified by showing that R'' is a generalized chordal bipartite graph, but the associated bipartite graph R' contains a chordless cycle of length $2k$. The same verification can be carried out for induced subgraphs of R.

Another example of a fragment R for which the one-or-same list problem is polynomial but where R'' is not even a generalized chordal bipartite graph has vertices x_i, y_i, z_i for $1 \leq i \leq 3$ and two additional vertices t, u, with edges tu, tx_i, ty_i, x_iy_j for $i \neq j$, y_iz_j for $i \neq j$, y_iu, z_iu, and a triangle with loops at the y_i.

9 Chordal bipartite graphs are symmetrically chordal extensible

Lemma 2 *Let G be a chordal graph, and let H be an induced subgraph of G. Then there is a perfect elimination ordering for G that first removes all vertices not in H if and only if for every connected component J of the subgraph of G induced by the vertices not in H, the vertices of H that are adjacent to vertices in J form a clique. In particular, this condition is met if H is induced by the vertices of a clique K and the vertices that are adjacent to all vertices in K.*

Proof. Suppose H meets the condition. In particular, if H is a clique, we may assume that H is a maximal clique, since any vertex can be eliminated from a clique. Let M be the subgraph of G induced by the vertices of a connected component J as above and the vertices of the clique H_J consisting of vertices in H adjacent to vertices in J.

We claim that H_J is not all of H. Otherwise H_J is a maximal clique, and J contains two vertices u and v whose neighbors in H_J form cliques H_u and H_v such that u is a vertex in J for which H_u is largest, and v is such that H_v is not a subset of H_u. Consider a shortest path from u to v in J given by $u = v_1, v_2, \ldots, v_k = v$. Let H_i be the clique of neighbors of v_i in H_J, where H_i may contain no vertices. Let r be the smallest i such that H_u contains a vertex not in H_{i+1}, and let x be such a vertex. Let s be the smallest $i \geq r+1$ such that H_i contains a vertex that is either

x or a vertex not in $H_i = H_u$, and let y be such a vertex. Then $x, v_r, v_{r+1}, \ldots, v_s, y$, possibly with $y = x$, form a chordless cylce of length at least four, contrary to chordality.

Since H_J is not all of H, we have that M is not all of G, and we may inductively infer that M has a perfect elimination ordering that first removes the vertices in J, leaving only the clique H_J. Applying such a perfect elimination ordering to each choice of J for G leaves only H, proving the if part.

If the condition is not met, then there are two vertices x and y in H that are not adjacent, but both x and y are each adjacent to some vertex in J. The subgraph M of G induced by x, y, and the vertices in J does not have a perfect elimination ordering that first removes all vertices in J, since the ordering cannot end up with two vertices that are not adjacent, proving the only if part.

Suppose H is induced by the vertices of a clique K and the vertices that are adjacent to all vertices in K. Suppose the condition is not met, and some J is such that there are two vertices x and y in H that are not adjacent, but both x and y are each adjacent to some vertex in J. Let p be a path $x, u_1, u_2, \ldots, u_t, y$ in G whose internal vertices u_i are in J, and with t smallest. For each vertex z in K, the vertices in the cycle $z, x, u_1, \ldots, u_t, y, z$ induce a subgraph of G that is not chordal, unless each u_i is adjacent to z. Then each u_i is adjacent to all vertices z in K, so u_i would have to be in H by the definition of H, a contradiction. □

Theorem 29 *Let $G = (V_1, V_2, E)$ be a chordal bipartite graph, and consider two subsets $U_1 \subseteq V_1$, $U_2 \subseteq V_2$ such that $U_1 \cup U_2$ induces a complete bipartite subgraph of G. Then there exists a chordal graph R on the vertices $V_1 \cup V_2$ such that the edges of R joining a vertex in V_1 and a vertex in V_2 are the edges of G, each V_i induces a connected subgraph of R, and the vertices $U_1 \cup U_2$ form a clique in R.*

Proof. Since G is chordal bipartite, G is also a generalized chordal bipartite graph. Thus there exists a chordal graph M on the vertices $V_1 \cup V_2$ such that the edges of M joining a vertex in V_1 and a vertex in V_2 are the edges of G, the subgraph of M induced by V_1 is a tree Q, and the edges of M joining two vertices x, y in V_2 are given by all pairs of vertices x, y that have a common neighbor in V_1.

We define R as follows. Let T be the set of vertices of G that are either in U_2 or adjacent in M to all vertices in U_2. Thus there is no edge from $V_1 \cap T$ to $V_2 \setminus T$, as any neighbor w of a vertex v of $V_1 \cap T$ in V_2 is adjacent to all of U_2 and thus in $V_2 \cap T$. Include in R all edges of G joining a vertex in V_1 and a vertex in V_2. Include in R all edges of G joining two vertices x, y in V_2 such that x and y are joined by a path $xu_1v_1u_2v_2 \ldots u_\ell y$ with the u_i, v_i preceeding x, y in the perfect elimination ordering, $u_i \in V_1 \setminus T, v_i \in V_2 \setminus T$ and $\ell \geq 1$. We denote this path by xWy. Furthermore include in R a clique on the vertices $V_1 \cap T$, and all edges joining a vertex in U_2 to a vertex in $V_2 \cap T$. Finally include the edges of Q in R.

We claim that M has a perfect elimination ordering that removes first the vertices not in T and only afterwards the vertices in T. We know by Lemma 2 that there is a perfect elimination order of M that removes the clique on U_2 last. Suppose a vertex u in $V_1 \cap T$ is removed just before a vertex v in $V_1 \setminus T$. As u is adjacent to all of U_2 and v is not adjacent to all of U_2, it follows that u and v cannot be adjacent and so v could be removed before u. Suppose a vertex u in $V_1 \cap T$ is removed just before a vertex v in $V_2 \setminus T$. As v is not in T, it follows that v cannot be adjacent to u, so v could be removed before u. Suppose a vertex u in $V_2 \cap T$ is removed just before a vertex v in $V_1 \setminus T$. As u is adjacent to all of U_2 and v is not adjacent to all of U_2, it follows that u and v cannot be adjacent and so v could be removed before u. Finally suppose a vertex u in $V_2 \cap T$ is removed just before a vertex v in $V_2 \setminus T$. As u is adjacent to all of u and v is not adjacent to all of U_2, it follows that u and v cannot be adjacent and so v could be removed before u.

Consider the perfect elimination ordering that first removes vertices not in T in M. This ordering can be used to remove vertices not in T in R, as whenever the neighborhood of such a vertex is a clique in M, it is also a clique in R. It thus remains to show that the subgraph R' of R induced by T is chordal.

Suppose R' has a chordless cycle C of length at least 4. As R' induces a clique on $V_1 \cap T$, it follows that C has at most two vertices in $V_1 \cap T$, and if it has two such vertices they must be consecutive in C. Suppose C has all vertices in $V_2 \cap T$. If we write C as $x_1x_2 \ldots x_kx_1$, $k \geq 4$, then each x_i is not in U_2, as otherwise it would be adjacent to all x_j, so each pair x_i, x_{i+1} and x_k, x_1 has a common path W_i or W_k in $V_1 \setminus T$, $V_2 \setminus T$, forming a cycle C' given by $x_1W_1x_2W_2 \ldots x_kW_kx_1$ in G. No two W_i can intersect or be adjacent to each other or to any x_i other than x_i, x_{i+1}, otherwise we would have a chod in C. Thus C' is a chordless cycle in G of length at least $2k \geq 8$, contrary to chordal bipartiteness of G.

Suppose C has exactly one vertex x_0 in $V_1 \cap T$. Write C as $x_0x_1x_2 \ldots x_kx_0$ with $k \geq 3$ and again find paths W_i in $V_1 \setminus T$, $V_2 \setminus T$ to obtain a chordless cycle $x_0x_1W_1x_2W_2 \ldots W_{k-1}x_kx_0$ in G of length at least $2k \geq 6$ in G, contrary to chordal bipartiteness of G. Finally suppose C has exactly two vertices x_0, x_1 in $V_1 \cap T$. Write C as $x_0x_1x_2 \ldots x_kx_0$ with $k \geq 3$. Find paths W_i in $V_1 \setminus T$, $V_2 \setminus T$ as before for $2 \leq i \leq k-1$, and select a vertex u_1 in U_2 that is not adjacent to the first vertex of W_2 to obtain a cycle C' in G given by $x_0u_1x_1x_2W_2 \ldots W_{k-1}x_kx_0$. If u_1 is not adjacent to any u_j in the W_j for $2 \leq j \leq k-1$, then C' is a chordless cycle of length $2k \geq 6$ in G. If u_1 has u_j in W_j as a first such neighbor for $3 \leq j \leq k-1$, then we obtain a chordlass cycle $u_1x_1x_2W_2 \ldots W'_ju_1$ of length at least 6 in G, where W'_j is the prefix of W_j up to u_j contrary to chordal bipartiteness. This completes the proof, as R is thus chordal. ∎

Applying Theorems 28 and 29, we obtain the following.

Corollary 4 *Let H be a mixed graph with the structure of Theorem 27. If every fragment R of H has a semi-fragment R'' that is chordal bipartite, then the OS-LIST H is polynomial.*

A graph whose associated bipartite graph is chordal bipartite has the structure of Theorem 27, so we get the following.

Corollary 5 *If H is a mixed graph whose associated bipartite graph is chordal bipartite, then OS-LIST H is polynomial.*

Mixed trees with connected loops have an associated bipartite graph that is chordal bipartite.

Corollary 6 *If H is a mixed tree with connected loops, then RET H and OS-LIST H are polynomial, while if the loops are not connected then both problems are NP-complete.*

References

[1] R. Brewster, T. Feder, P. Hell, J. Huang and G. MacGillivray. Near-unanimity functions and varieties of graphs.

[2] T. Feder and P. Hell, List homomorphisms to reflexive graphs J. Comb. Theory Series B 72-2 (1998) 236-250.

[3] T. Feder, P. Hell, and J. Huang, List homomorphisms and circular arc graphs. Combinatorica 19-4 (1999) 487-505.

[4] T. Feder, P. Hell, and J. Huang, Bi-arc graphs and the complexity of list homomorphisms. J. Graph Theory 42 (1999) 61-80.

[5] 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) 57-104.

[6] P. Hell, J. Nešetřil, and X. Zhu, Duality and polynomial testing of tree homomorphisms. Transactions of the AMS.

[7] Larose CSP special near-unanimity, first order definable, finitely many minimal obstructions.

[8] Larose partial order near-unanimity, finitely many minimal obstruction.

[9] Larose partial order near-unanimity has width one.

[10] Larose reflexive near-unanimity has special near-unanimity.

[11] Nesetril first order and width one.