Contracting a chordal graph to a split graph or a tree*

Petr A. Golovach[1], Marcin Kamiński[2], and Daniël Paulusma[1]

[1] School of Engineering and Computing Sciences, Durham University, Science Laboratories, South Road, Durham DH1 3LE, United Kingdom
{petr.golovach,daniel.paulusma}@durham.ac.uk
[2] Département d'Informatique, Université Libre de Bruxelles, Belgium
marcin.kaminski@ulb.ac.be

Abstract. The problems Contractibility and Induced Minor are to test whether a graph G contains a graph H as a contraction or as an induced minor, respectively. We show that these two problems can be solved in $|V_G|^{f(|V_H|)}$ time if G is a chordal input graph and H is a split graph or a tree. In contrast, we show that containment relations extending Subgraph Isomorphism can be solved in linear time if G is a chordal input graph and H is an arbitrary graph not part of the input.

1 Introduction

There are several natural and elementary algorithmic problems to test whether the structure of some graph H shows up as a *pattern* within the structure of another graph G. We focus on one such problem in particular, namely whether one graph contains some other graph as a contraction. Before we give a survey of existing work and present our results, we first state some basic terminology.

We consider undirected finite graphs that have no loops and no multiple edges. We denote the vertex set and edge set of a graph G by V_G and E_G, respectively. If no confusion is possible, we may omit the subscripts. We refer to Diestel [7] for any undefined graph terminology. Let $e = uv$ be an edge in a graph G. The *edge contraction* of e removes u and v from G, and replaces them by a new vertex adjacent to precisely those vertices to which u or v were adjacent. A graph H is a *minor* of a graph G if H can be obtained from G by a sequence of vertex deletions, edge deletions, and edge contractions. If only vertex deletions and edge contractions are allowed, then H is an *induced minor* of G. If only edge contractions are allowed, then H is a *contraction* of G. If only vertex deletions and edge deletions are allowed, then H is a *subgraph* of G. The corresponding decision problems are called Minor, Induced Minor, Contractibility, and Subgraph Isomorphism, respectively.

Matoušek and Thomas [21] showed that Contractibility, Induced Minor and Minor are NP-complete even on ordered input pairs (G, H) where G

* This work is supported by EPSRC (EP/G043434/1).

and H are trees of bounded diameter, or G and H are trees with at most one vertex of degree more than 5. If H is a cycle on $|V_G|$ vertices, then Subgraph Isomorphism is equivalent to asking whether G is Hamiltonian. This is an NP-complete problem even when G is restricted to be chordal bipartite, as shown by Müller [22]. It is therefore natural to fix the graph H in an ordered input pair (G, H) and consider only the graph G to be part of the input. We indicate this by adding "H-" to the names of the decision problems.

A celebrated result by Robertson and Seymour [24] states that H-Minor can be solved in cubic time. It is straightforward that H-Subgraph can be solved in polynomial time for any fixed graph H. The computational complexity classifications of the problems H-Induced Minor and H-Contractibility are still open, although many partial results are known. Fellows, Kratochvíl, Middendorf, and Pfeiffer [9] gave both polynomial-time solvable and NP-complete cases for the H-Induced Minor problem. The smallest known NP-complete case is a graph H on 68 vertices [9]. A number of polynomial-time solvable and NP-complete cases for the H-Contractibility problem can be found in a series of papers started by Brouwer and Veldman [5] and followed by Levin, Paulusma and Woeginger [19, 20] and van 't Hof et al. [17]. The smallest NP-complete cases are when H is a 4-vertex path or a 4-vertex cycle [5]. Because some of the open cases for both problems are notoriously difficult, special graph classes have been studied in the literature. Fellows, Kratochvíl, Middendorf, and Pfeiffer [9] showed that for every fixed graph H, the H-Induced Minor problem can be solved in polynomial time on planar graphs. Also the H-Contractibility problem can be solved in polynomial time for every fixed H on this graph class [18].

A graph G is a *split graph* if G has a *split partition*, which is a partition of its vertex set into a clique C_G and an independent set I_G. Split graphs were introduced by Foldes and Hammer [10] in 1977 and have been extensively studied since then; see e.g. the monographs of Brandstädt, Le and Spinrad [4], or Golumbic [14]. Belmonte, Heggernes, and van 't Hof [1] showed that Contractibility is NP-complete for ordered input pairs (G, H) where G is a split graph and H is a split graph of a special type, namely a threshold graph. They also showed that for every fixed graph H, the H-Contractibility problem can be solved in polynomial time for split graphs. As a matter of fact, H may be assumed to be a split graph in this result, because split graphs are closed under taking contractions. Golovach et al. [13] showed that Minor and Induced Minor are NP-complete for ordered input pairs (G, H) where G and H are split graphs. They also showed that Contractibility and Induced Minor are W[1]-hard for such input pairs (G, H) when parameterized by $|V_H|$. Hence, it is unlikely that these two problems can be solved in $|V_G|^{O(1)}$ time for such input pairs (G, H) with the constant in the exponent independent of H. The same authors [13] showed that H-Induced Minor is polynomial-time solvable on split graphs for any fixed graph H. Because split graphs are closed under taking induced minors, also in this result H may be assumed to be a split graph.

A graph is called *chordal* (or *triangulated*) if it contains no induced cycle on at least four vertices; note that every split graph is chordal. Heggernes, van 't

Hof, Lévěque, and Paul [16] showed that H-Contractibility can be solved in polynomial time for chordal graphs when H is an a fixed path by checking whether the diameter of the chordal input graph is greater than or equal to the length of the fixed path. We observe that testing whether a graph G contains a fixed path H as an induced minor is equivalent to testing whether G contains H as an induced subgraph. This means that H-Induced Minor is polynomial-time solvable for general graphs if H is a fixed path; this problem is open if H is a fixed tree.

Our Results. We extend the aforementioned results of Belmonte, Heggernes, and van 't Hof [1] and Golovach et al. [13] by showing that H-Contractibility and H-Induced Minor can be solved in polynomial time on chordal graphs for any fixed split graph H. We also show that H-Contractibility and H-Induced Minor are polynomial-time solvable for chordal graphs when H is any fixed tree. This extends the aforementioned result of Heggernes, van 't Hof, Lévěque, and Paul [16]. In contrast to the W[1]-hardness of Contractibility and Induced Minor for split graphs [13], we show that the problems Subgraph Isomorphism, Minor and the related problems Topological Minor and Immersion, which we define later, can be solved in linear time if G is a chordal graph and H is an arbitrary fixed graph not part of the input.

Future Work. We leave it as an open problem whether H-Contractibility and H-Induced Minor can be solved in polynomial time on chordal graphs when H is any fixed chordal graph not necessarily a split graph or a tree.

2 Preliminaries

Let $G = (V, E)$ be a graph. A subset $U \subseteq V$ is a *clique* if there is an edge in G between any two vertices of U, and U is an *independent set* if there is no edge in G between any two vertices of U. We write $G[U]$ to denote the subgraph of G *induced* by $U \subseteq V$, i.e., the graph on vertex set U and an edge between any two vertices whenever there is an edge between them in G. Two sets $U, U' \subseteq V$ are called *adjacent* if there exist vertices $u \in U$ and $u' \in U'$ such that $uu' \in E$. A vertex v is a *neighbor* of u if $uv \in E$. The *degree* $d_G(u)$ of a vertex u is its number of neighbors. A set $U \subset V$ is a *cut-set* if $G - U$ is disconnected; if $U = \{u\}$, then u is called a *cut-vertex*. A vertex $v \in V$ is called *simplicial* if its neighbors in G form a clique.

For our proofs the following global structure is useful. Let G and H be two graphs. An *H-witness structure* $\mathcal{W}$ is a vertex partition of G into $|V_H|$ (nonempty) sets $W(x)$ called *H-witness bags*, such that

(i) each $W(x)$ induces a connected subgraph of G;
(ii) for all $x, y \in V_H$ with $x \neq y$, bags $W(x)$ and $W(y)$ are adjacent in G if x and y are adjacent in H;
(iii) for all $x, y \in V_H$ with $x \neq y$, bags $W(x)$ and $W(y)$ are adjacent in G *only* if x and y are adjacent in H.

By contracting all bags to singletons we observe that H is a contraction of G if and only if G has an H-witness structure such that conditions (i)-(iii) hold. Because every vertex $u \in V_G$ is in at most one bag, we can define $W_u = W(x)$ if u is in $W(x)$. We also use the shorthand notation $W(X) = \bigcup_{x \in X} W(x)$ for some $X \subseteq V_H$. We note that G may have more than one H-witness structure with respect to the same containment relation. Let H be an induced subgraph of G. We say that G has a *subgraph contraction* to H if G has an H-witness structure $\mathcal{W}$ such that $x \in W(x)$ for all $x \in V_H$. We also say that a vertex $u \in V_G \setminus V_H$ is *contracted to* $x \in V_H$ if $u \in W(x)$.

A clique tree $\mathcal{T}_G$ of a (connected) graph G is a tree that has as vertices the maximal cliques of G and has edges such that each graph induced by those cliques that contain a particular vertex of G is a subtree. We let $\mathcal{K}_G$ denote the set of all maximal cliques of G. The following three lemmas are well-known and useful; from now on we implicitly assume that we can compute a clique tree of a chordal graph in linear time whenever we need such a tree for our algorithms.

Lemma 1 ([12]). *A connected graph is chordal if and only if it has a clique tree.*

Lemma 2 ([23]). *Let $G = (V, E)$ be a chordal graph. Then $\sum_{K \in \mathcal{K}_G} |K| = O(|V| + |E|)$.*

Lemma 3 ([3, 11]). *A clique tree of a connected chordal graph $G = (V, E)$ can be constructed in $O(|V| + |E|)$ time.*

Let $G = (V, E)$ be a chordal graph. We refer to a set $K \in \mathcal{K}_G$ as a *node* of $\mathcal{T}_G$. We define the notions *root node*, *parent node*, *child node* and *leaf node* of a clique tree similar to the notions root, parent, child and leaf of a "normal" tree. If the bag $K_r \in \mathcal{K}_G$ is the root node of $\mathcal{T}_G$, then we say that T is *rooted at* K_r. A *descendant* of a node K is a node K^* such that K lies on the (unique) path from K^* to the root node K_r in $\mathcal{T}_G$; note that each node is its own descendant. Every node $K \neq K_r$ of a clique tree $\mathcal{T}_G$ has exactly one parent node K' in $\mathcal{T}_G$. We say that a vertex $v \in K$ is *given to the parent node* K' if $v \in K \cap K'$, i.e., if v is both in the child node K and in the parent node K'. We say that vertex $v \in K$ *stays behind* if $v \in K \setminus K'$, i.e., if v is in the child node K but not in the parent node K'. Bernstein and Goodman [2] showed that a tree $\mathcal{T}$ with vertex set $\mathcal{K}_G$ is a clique tree of G if and only if $\mathcal{T}$ is a maximum weight spanning tree of the *clique graph* $C(G)$ of G; this is the weighted graph that has as vertices the maximal cliques of G and that has an edge K_1K_2 with weight $|K_1 \cap K_2|$ whenever $K_1 \cap K_2 \neq \emptyset$. This leads to the following observation that we will implicitly use in the proofs of our results.

Observation 1 *Let G be a connected chordal graph with at least two maximal cliques. Let $\mathcal{T}_G$ be a clique tree of G rooted at K_r. At least one vertex of any node $K \neq K_r$ of T is given to the parent node of K and at least one vertex stays behind. Moreover, $|K| \geq 2$ for all $K \in \mathcal{K}_G$.*

3 Contracting to Split Graphs

A split graph with split partition (C, I) is called *maximal* if C is a maximal clique of the split graph. This means that there is no vertex in I adjacent to all vertices in C.

Throughout this section, we assume that G denotes a chordal graph with set of maximal cliques $\mathcal{K}_G$ and that H denotes a split graph with a maximal split partition (C_H, I_H), where $C_H = \{x_1, \ldots, x_p\}$ and $I_H = \{y_1, \ldots, y_q\}$. If G has an H-witness structure then we call the bags corresponding to the vertices in C_H and I_H *clique bags* and *independent bags*, respectively.

If $q = 0$, then G has H as a contraction if and only if G has H as a minor, and we can use Robertson and Seymour's theorem [24] to test this in polynomial time. Hence, we assume that $p \geq 1$ and $q \geq 1$. Because H is connected, we may assume without loss of generality that G is connected.

We start with the following observation, which follows from the definition of an H-witness structure.

Observation 2 *If G contains H as a contraction, then no clique of G has two vertices belonging to two non-adjacent bags of some H-witness structure of G.*

The following lemma is crucial.

Lemma 4. *If G contains H as a contraction, then G has an H-witness structure $\mathcal{W}$ such that for every maximal clique K of G exactly one of the following statements hold:*

(i) $K \subseteq W(C_H)$*, or*
(ii) $K \subseteq W(y_j)$ *for some* $y_j \in I_H$*, or*
(iii) *there is a vertex* $u \in K$ *with* $K \setminus \{u\} \subseteq W(C_H)$ *and* $u \in W(y_j)$ *for some* $y_j \in I_H$*.*

Moreover, for every $y_j \in I_H$, there exists at least one maximal clique K of G that contains a vertex u with $u \in W(y_j)$ and $K \setminus \{u\} \subseteq W(C_H)$.

Proof. Suppose that G contains H as a contraction. Let $\mathcal{W}$ be an H-witness structure of G such that $W(C_H)$ is maximal. Let K be a maximal clique of G. Suppose that (i) and (ii) do not hold. By Observation 2, we then find that K contains three vertices u_1, u_2 and v, such that u_1 and u_2 belong to some independent bag $W(y_j)$ and v belongs to $W(C_H)$.

Let U_1 be the component of $G[W(y_j) \setminus \{u_2\}]$ that contains u_1, and let U_2 be the component of $G[W(y_j) \setminus \{u_1\}]$ that contains u_2. If U_2 is adjacent to every clique bag that is adjacent to U_1, then we move U_1 from $W(y_j)$ to the clique bag that contains v. We do the same with U_2 in the case that U_1 is adjacent to every clique bag that is adjacent to U_2. In both cases, the maximality of $W(C_H)$ is violated. Hence, there exists a clique bag $W(x_h)$ that is adjacent to U_1 but not to U_2, and a clique bag $W(x_i)$ that is adjacent to U_2 but not to U_1. We let s be a vertex in U_1 that has a neighbor t in $W(x_h)$ and s' be a vertex in U_2 that has a neighbor t' in $W(x_i)$. Here, we choose s as close as possible to u_1 and s' as close as possible to u_2; note that s and s' might be equal to u_1 and u_2, respectively. Let P be a shortest path from s to s' in $G[W(y_j)]$. Note that P passes through u_1 and u_2. By definition, s is the only vertex on P that has a neighbor, namely t, in $W(x_h)$, and s' is the only vertex on P that has a neighbor, namely t', in $W(x_i)$. In $G[W(x_h) \cup W(x_i)]$, we choose a shortest path Q from t to t'. Then the paths P and Q together with the two edges st and $s't'$ form an induced cycle in G on at least four vertices. This is not possible, because G is chordal.

To prove the second statement of Lemma 4, consider a vertex $y_j \in I_H$. Let $x_i \in C_H$ be a neighbor of y_j. Then G must contain an edge tu with $t \in W(x_i)$ and $u \in W(y_j)$. Let K be a maximal clique of G that contains t and u. Because $\mathcal{W}$ satisfies the first statement of Lemma 4, we find that $K \setminus \{u\} \subseteq W(C_H)$, as desired. Hence, we have proven Lemma 4. $\square$

We say that an H-witness structure $\mathcal{W}$ that satisfies Lemma 4 is *elementary*. We call the maximal clique K and the vertex u in the second statement of Lemma 4 a *y_j-characteristic pair* for $\mathcal{W}$.

We are now ready to describe our algorithm. We start with some branching as described in the following lemma, the proof of which can be found in Appendix A.

Lemma 5. *We can obtain in polynomial time a set $\mathcal{G}$ of graphs, such that G contains H as a contraction if and only if there exists a graph $G' \in \mathcal{G}$ that has an H-witness structure $\mathcal{W}'$ with $W'(y_j) = \{u_j(G')\}$ for $j = 1, \ldots, q$, where $u_1(G'), \ldots, u_q(G')$ are q specified mutually non-adjacent vertices of G' that are not cut-vertices of G'.*

Suppose that we have obtained a set $\mathcal{G}$ that satisfies Lemma 5. Because $\mathcal{G}$ has polynomial size, we process each graph in it one by one. For simplicity, we will denote such a graph by G again and its set of q specified vertices by $u_1, \ldots, u_q$.

Let G^* be the graph obtained from G after removing $u_1, \ldots, u_q$. We make the following observation, which follows from the property of G that the intersection of two intersecting maximal cliques contain a vertex from $V_G \setminus \{u_1, \ldots, u_q\}$ as the vertices $u_1, \ldots, u_q$ are no cut-vertices and form an independent set in G.

Observation 3 *The graph G^* is a connected chordal graph.*

What is left to decide is if and how the vertices of G^* can be distributed over the clique bags of a witness structure $\mathcal{W}$ of G with $W(y_j) = \{u_j\}$ for $j = 1, \ldots, q$. In order to do this we follow a dynamic programming approach over a clique tree of G. For this purpose we must first decide how to root this tree, and in order to do that we need the following lemma.

Lemma 6. *Let C be a cut-set of G^*. If G has an H-witness structure $\mathcal{W}$ with $W(y_j) = \{u_j\}$ for $j = 1, \ldots, q$, then for all pairs of vertices $v_1, v_2 \in V_{G^*}$ that are in two different components of $G^* - C$, there exists a vertex $t \in C$ such that $W_{v_1} = W_t$ or $W_{v_2} = W_t$.*

Proof. Suppose that there exist two vertices $v_1, v_2 \in V_{G^*}$ that are in two different components of $G^* - C$ such that there is no vertex $t \in C$ with $W_{v_1} = W_t$ or $W_{v_2} = W_t$. Let $W_{v_1} = W(x_h)$ and $W_{v_2} = W(x_i)$. Then either $G[W(x_h)]$ is disconnected if $h = i$, or $W(x_h)$ and $W(x_i)$ are not adjacent if $h \neq i$. Both cases are not possible. □

Lemma 6 helps us to deduce a useful property for our dynamic programming; we describe this property in Lemma 7.

Lemma 7. *If G has an H-witness structure $\mathcal{W}$ with $W(y_j) = \{u_j\}$ for $j = 1, \ldots, q$, then there exists a node K_r^* such that $\mathcal{T}_{G^*}$ can be rooted at K_r^* with the following property valid for every parent node K_p^* with child node K_c^*: for all $v \in K_c^* \setminus K_p^*$ there exists a vertex $t \in K_c^* \cap K_p^*$ with $W_t = W_v$.*

Proof. Let K be a node of $\mathcal{T}_{G^*}$. Suppose that the property in the statement of the lemma does not hold if we choose K to be the root node. Then there exists a node K_1 that has a child node K_2 such that $K_2 \setminus K_1$ has a vertex v for which there does not exist a vertex $t \in K_1 \cap K_2$ with $W_t = W_v$. We apply Lemma 6 for $C = K_1 \cap K_2$ and find that for every vertex v' in the component of $G^* - C$ that contains $K_2 \setminus K_1$ there exists a vertex $t' \in K_1 \cap K_2$ with $W_{t'} = W_{v'}$. We now choose as new root the node K_2. If K_2 satisfies the property in the statement of the lemma, then we are done. Otherwise, there exists a node K_3 with a child node K_4 as before. However, K_4 cannot be in the component of $G^* - C$ that contains the vertices of $K_1 \setminus K_2$. Hence, as G^* is a finite graph, repeatedly applying the same argument, eventually yields a root node that satisfies the statement of the lemma. □

From now we assume that $\mathcal{T}_{G^*}$ is rooted in a node K_r^* in such a way that the property described in Lemma 7 holds if G contains H as a contraction. We may do this, because we will consider if necessary all nodes of $\mathcal{T}_{G^*}$ to be the root node, and consequently repeat the algorithm a number of times. This number is polynomially bounded, because the total number of different nodes in $\mathcal{T}_{G^*}$ is polynomially bounded due to Lemma 2. Our next aim is to delete simplicial vertices of G that are in $V_{G^*} \setminus K_r^*$.

Lemma 8. *Let $v \in V_{G^*} \setminus K_r^*$ be a simplicial vertex of G. If G has an H-witness structure for G with $W(y_j) = \{u_j\}$ for $j = 1, \ldots, q$ where $v \in W(x_i)$ is adjacent to some vertex of $W(x_j)$ for some $x_i, x_j \in C_H$, then G has an H-witness structure $\mathcal{W}'$ with*

(i) $W'(z) = W(z)$ *for* $z \in V_H \setminus \{x_i, x_j\}$, *and*
(ii) $W'(x_i) = W(x_i) \setminus \{v\}$, *and*
(iii) $W'(x_j) = W(x_j) \cup \{v\}$.

Proof. Because $v \notin K_r^*$, we find that $W'(x_i) \neq \emptyset$ due to Lemma 7. Because v is simplicial, $W'(x_i)$ is connected. Because v is adjacent to a vertex from $W(x_j)$, we find that $W'(x_j)$ is connected. Because v is a simplicial vertex of G, any neighbor of v in $\{u_1, \ldots, u_q\}$ is adjacent to a vertex of $W'(x_i)$ and to a vertex of $W(x_j)$. □

Lemma 8 implies that simplicial vertices in $V_{G^*} \setminus K_r^*$ can be included in any adjacent bag. This means that we can exclude them from the graph recursively, and from now we assume that G has no simplicial vertices in $V_{G^*} \setminus K_r^*$.

We now apply dynamic programming over $\mathcal{T}_{G^*}$. Our algorithm returns `Yes` if G has an H-witness structure $\mathcal{W}$ with $W(y_j) = \{u_j\}$ for $j = 1, \ldots, q$ that satisfies the property given in Lemma 7; It returns `No` otherwise. We refer to Appendix B for the details of this algorithm, a correctness proof and a running time analysis, which leads us to the main result of this section.

Theorem 1. *For any fixed split graph H, the H-*Contractibility *problem can be solved in polynomial time for chordal graphs.*

4 Contracting to Trees

Throughout this section, we assume that G denotes a chordal graph with set of maximal cliques $\mathcal{K}_G$ and that H denotes a tree with leaves $\{z_1, \ldots, z_r\}$. We also let $\mathcal{T}_G$ denote a clique tree of G. Due to Theorem 1, we may assume that H is not a split graph. Hence, H has at least four vertices. We will present an algorithm that decides in $|V(G)|^{|O(|V_H|)}$ time whether G contains H as a contraction.

If G has an H-witness structure $\mathcal{W}$ then we call the bags $W(z_i)$ the *leaf bags* of $\mathcal{W}$. We define *parent bag* and *child bag* of $\mathcal{W}$ analogously. We start with the following lemma.

Lemma 9. *If G contains H as a contraction, then G has an H-witness structure $\mathcal{W}$ with $W(z_i) = \{u_i\}$, $i = 1, \ldots, r$ for some set of vertices $u_1, \ldots, u_r$.*

Proof. Suppose that G contains H as a contraction. Let $\mathcal{W}$ be an H-witness structure of G. Suppose that $|W(z_i)| \geq 2$ for some $1 \leq i \leq r$. Let y_i be the parent of z_i in H. We choose a vertex $u_i \in W(z_i)$ that is not a cut-vertex of $G[W(z_i)]$ and move all vertices in $W(z_i) \setminus \{u_i\}$ to $W(y_i)$. This leads to a new H-witness structure of G, in which the leaf bag corresponding to z_i only contains u_i. If necessary, we apply this operation on every other leaf bag that contains more than one vertex. $\square$

We call the vertices $u_1, \ldots, u_r$ in Lemma 9 the *leaf bag vertices* of the witness structure $\mathcal{W}$ and call $\mathcal{W}$ a *simple* witness structure. We can now show the following lemma, the proof of which can be found in Appendix C.

Lemma 10. *We can obtain in polynomial time a set $\mathcal{G}$ of graphs, such that G contains H as a contraction if and only if there exists a graph $G' \in \mathcal{G}$ that has a simple H-witness structure $\mathcal{W}'$ with as leaf bag vertices q specified vertices $u_1(G'), \ldots, u_r(G')$ that are of degree one in G' and that together form the set of all simplicial vertices of G'.*

From now on suppose that we have obtained a set $\mathcal{G}$ that satisfies Lemma 10. Because $\mathcal{G}$ has polynomial size, we can process each graph in it. Therefore, we may without loss of generality assume that we have a connected chordal input graph with a set of q specified vertices $u_1, \dots, u_q$ that are of degree one and that together form the set of all simplicial vertices of the graph. For simplicity, we will denote this chordal input graph by G again.

If G allows an H-witness structure $\mathcal{W}$, then we call a maximal clique K of G a *vertex clique* of $\mathcal{W}$ if K contains vertices from exactly one witness bag from $\mathcal{W}$, and we call K an *edge clique* of $\mathcal{W}$ if K contains vertices from exactly two (adjacent) witness bags from $\mathcal{W}$. For a vertex $x \in V_H$, we call a vertex-clique an x-vertex-clique if it is a subset of $W(x)$. For an edge $xy \in E_H$, we call an edge-clique an xy-edge-clique if it contains one or more vertices of $W(x)$ and one or more vertices of $W(y)$.

We make two useful observations. The first observation holds, because H is a tree. The second observation follows from Observation 4 and the property that every bag of a witness structure is connected.

Observation 4 *If G contains H as a contraction, then every maximal clique of G is either a vertex clique or an edge clique in every H-witness structure of G.*

Observation 5 *If G contains H as a contraction, then for all $xy \in E_H$ the xy-edge cliques in every H-witness structure of G are the nodes of a connected subtree of $\mathcal{T}_G$.*

We are now ready to show our last lemma. Recall that G contains exactly r vertices $u_1, \dots, u_r$ of degree 1 and that G contains no other simplicial vertices.

Lemma 11. *There exists a polynomial-time algorithm that tests whether G has an H-witness structure $\mathcal{W}$ such that $W(y_i) = \{u_i\}$ for $i = 1, \dots, r$.*

Proof. By Observation 4, all maximal cliques of G are either vertex-cliques or edge-cliques for any H-witness structure (if it exists). By Observation 11, the xy-edge cliques are in that case the nodes of a connected subtree $\mathcal{T}_{xy}$ of $\mathcal{T}_G$. We do as follows. For every $xy \in E_H$, we guess the nodes of $\mathcal{T}_{xy}$. This leads to a collection of $|E_H|$ subtrees that are to correspond to the subtrees $\mathcal{T}_{xy}$ of $\mathcal{T}_G$. Afterwards, we must perform a number of checks to see if such a family of subtrees is a “good” guess, i.e., leads to an H-witness structure $\mathcal{W}$ with the required properties.

Firstly, we must check if there is no node of $\mathcal{T}_G$ that is contained in two different subtrees. If so, then we discard our guessed family of subtrees. Otherwise we continue as follows. Because we have chosen nodes to correspond to all edge-cliques, all other nodes of $\mathcal{T}_G$ must correspond to the vertex-cliques. We may therefore contract these nodes to single vertices in G. Because single vertices are not maximal cliques, this leads to a new graph G^* with clique tree $\mathcal{T}_{G^*}$, the nodes of which are to correspond to the edge-cliques.

If there exist two maximal cliques K and K' of G^* with $K \cap K' \neq \emptyset$ that are nodes of two subtrees that are to correspond to subtrees $\mathcal{T}_{xy}$ and $\mathcal{T}_{x'y'}$ with $\{x, y\} \cap \{x', y'\} = \emptyset$, then we discard our guessed family of subtrees; every vertex in $K \cap K'$ must go to $W(x) \cup W(y)$ and $W(x') \cup W(y')$ simultaneously and this is not possible. Suppose that such a pair of maximal cliques does not exist. Then for every intersecting pair of maximal cliques from two different guessed subtrees corresponding to subtrees T_{xy} and $T_{x'y'}$ we have without loss of generality $x = x'$. This means that the vertices in this intersection must go into $W(x)$. Because of this, we contract the edges in the intersection of two such cliques into one vertex which we put in a new set A_x. In the end we must check for every $xy \in E_H$, if the vertices of subtree T'_{xy} can be partitioned into two sets B_1 and B_2 such that (i) $A_x \subseteq B_1$, (ii) $A_y \subseteq B_2$ and (iii) $G[B_1]$ and $G[B_2]$ are both connected. If so, then we have our desired H-witness structure of G. Otherwise we must consider some other family of guessed subtrees.

This algorithm runs in polynomial time for the following reasons. For every $xy \in E_H$, we actually only have to guess the nodes in $\mathcal{T}_G$ that correspond to the leaves of $\mathcal{T}_{xy}$. Because every leaf node of $\mathcal{T}_G$ has a simplicial vertex, $\mathcal{T}_G$ contains exactly r leaf nodes. This means that the number of nodes we must guess for $\mathcal{T}_G$ is at most r, which is a constant because we assume that r is fixed. We can process each set of guessed subtrees in polynomial time as well. In particular, we can check if A_x and A_y can be made connected in a subtree $\mathcal{T}_G$ by applying Robertson and Seymour's algorithm [24] for this problem, which is called the 2-Disjoint Connected Subgraphs problem. Their algorithm runs in polynomial time as long as the total number of specified vertices is bounded. This is the case in our setting, because the number of leaves of every guessed subtree is bounded and the intersection with other subtrees $\mathcal{T}_{x'y'}$ must always contain vertices from the leaves or the root of the guessed subtrees. This completes the proof of Lemma 11. □

By Lemmas 10 and 11 we obtain the main result of this section.

Theorem 2. *For any fixed tree H, the H-Contractibility problem can be solved in polynomial time for chordal graphs.*

5 Induced Minors and Extensions of Subgraphs

We let $P_1 \bowtie G$ denote the graph obtained from a graph G after adding a new vertex and making it adjacent to all vertices of G.

Lemma 12 ([17]). *Let H and G be two graphs. Then G has H as an induced minor if and only if $P_1 \bowtie G$ is $(P_1 \bowtie H)$-contractible.*

We observe that $P_1 \bowtie G$ is a chordal graph if G is chordal. Similarly, $P_1 \bowtie G$ is a split graph if G is a split graph. Then, combining Lemma 12 with Theorem 1 yields the following result.

Theorem 3. *For any fixed split graph H, the H-Induced Minor problem can be solved in polynomial time for chordal graphs.*

We also have the following result, which can be proven by using the same arguments as the proof of Theorem 2 after making a minor modification: if there exist two maximal cliques K and K' of the graph G^* in the proof of Lemma 11 with $K \cap K' \neq \emptyset$ that are nodes of two subtrees that are to correspond to subtrees $\mathcal{T}_{xy}$ and $\mathcal{T}_{x'y'}$ with $\{x,y\} \cap \{x',y'\} = \emptyset$, then we remove these vertices instead of discarding the family of guessed subtrees.

Theorem 4. *For any fixed tree H, the H-*Induced Minor *problem can be solved in polynomial time for chordal graphs.*

We conclude the paper with an observation that containment relations extending Subgraph Isomorphism can be decided in linear time on a chordal input graph G if the graph H is not considered to be part of the input. The H-Subgraph Isomorphism problem most likely cannot be solved in $|V_G|^{O(1)}$ time for a general graph G when the constant in the exponent is independent of H, as shown by Downey and Fellows [8]. Besides minors, we consider the following of such containment relations. A graph G contains H as a *topological minor* if H can be obtained from the input graph by deleting vertices and edges and contracting edges incident with a vertex of degree 2. Given a (not necessarily induced) path abc on three vertices, a *lift* is to remove the edges ab and bc and add the edge ac (if is not already present in the graph). A graph G contains H as a *immersion* if H can be obtained from G by vertex and edge deletions and lifts. The corresponding decision problems are called Topological Minor and Immersion. For any fixed graph H, both H-Topological Minor and H-Immersion have recently been shown to be solvable in $O(|V(G)|^3)$ time by Grohe, Kawarabayashi, Marx and Wollan [15].

Theorem 5. *For any fixed graph H, the four problems H-*Subgraph Isomorphism*, H-*Minor*, H-*Topological Minor*, and H-*Immersion *can be solved in $O(|V| + |E|)$ time for chordal graphs.*

Proof. Let G be the input chordal graph. We construct a clique tree $\mathcal{T}_G$ of G in linear time due to Lemma 3. In the same time, we can find a largest maximal clique in $\mathcal{T}_G$. We denote its size by c. If $c \geq |V_H|$, then any largest clique contains H as a subgraph and thus as a minor, topological minor, and immersion. Hence, G contains H as a subgraph, minor, topological minor, and immersion. Otherwise, the size of all cliques in $\mathcal{T}_G$ is at most c and $\mathcal{T}_G$ is a tree-decomposition of G of width $c-1$. All four relations H-Subgraph Isomorphism, H-Minor, H-Topological Minor, and H-Immersion can be decided in $O(|V| + |E|)$ time for graphs of bounded treewidth [6, 15].

References

1. R. Belmonte, P. Heggernes, and P. van 't Hof. Edge contractions in subclasses of chordal graphs. In: Proceedings of the 8th Annual Conference on Theory and Applications of Models of Computation (TAMC 2011). *Lecture Notes in Computer Science*, to appear.
2. P.A. Bernstein and N. Goodman. Power of natural semijoins. *SIAM Journal on Computing 10* (1981) 751–771.

3. J.R.S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. *Graph Theory and Sparse Matrix Computation 56* (1993) 1–29.
4. A. Brandstädt, V.B. Le, and J. Spinrad. Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications, 1999.
5. A.E. Brouwer and H.J. Veldman. Contractibility and NP-completeness. *Journal of Graph Theory*, 11: 71–79, 1987.
6. B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.
7. R. Diestel. *Graph Theory*. Springer-Verlag, Electronic Edition, 2005.
8. R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W[1]. Theoretical Computer Science 141:109–131, 1995.
9. M.R. Fellows, J. Kratochvíl, M. Middendorf, and F. Pfeiffer. The Complexity of Induced Minors and Related Problems. *Algorithmica*, 13: 266–282, 1995.
10. S. Foldes, P. L. Hammer. Split graphs. Proceedings of the Eighth Southeastern Conference on Combinatorics, Graph Theory and Computing (Louisiana State Univ., Baton Rouge, La., 1977), pp. 311315. *Congressus Numerantium*, No. XIX, Utilitas Math., Winnipeg, Man., 1977.
11. P. Galinier, M. Habib, and C. Paul. Chordal Graphs and Their Clique Graphs. In *21st International Workshop on Graph-Theoretic Concepts in Computer Science (WG 1995)* Lecture Notes in Computer Science 1017 (1995) 358–371.
12. F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B 16* (1974) 47–56.
13. P.A. Golovach, M. Kaminski, D. Paulusma and D.M. Thilikos, Containment relations in split graphs, manuscript.
14. M.C. Golumbic. Algorithmic Graph Theory and Perfect Graphs. Annals of Discrete Mathematics 57, Elsevier, 2004.
15. M. Grohe. K. Kawarabayashi, D. Marx and P. Wollan, Finding topological subgraphs is fixed-parameter tractable. To appear in *Symposium on Theory of Computing, STOC'11, 2011*.
16. P. Heggernes, P. van 't Hof, B. Lévěque, and C. Paul. Contracting chordal graphs and bipartite graphs to paths and trees. In: Proceedings of LAGOS 2011, Electronic Notes in Discrete Mathematics, to appear.
17. P. van 't Hof, M. Kamiński, D. Paulusma, S. Szeider and D.M. Thilikos. On graph contractions and induced minors. *Discrete Applied Mathematics*, to appear.
18. M. Kamiński, D. Paulusma and D.M. Thilikos. Contractions of planar graphs in polynomial time. In: Proceedings of the 18th Annual European Symposium on Algorithms (ESA 2010). *Lecture Notes in Computer Science 6346*, 122–133, 2010.
19. A. Levin, D. Paulusma, and G.J. Woeginger. The computational complexity of graph contractions I: polynomially solvable and NP-complete cases. *Networks*, 51: 178–189, 2008.
20. A. Levin, D. Paulusma, and G.J. Woeginger. The computational complexity of graph contractions II: two tough polynomially solvable cases. *Networks*, 52: 32–56, 2008.
21. J. Matoušek and R. Thomas. On the complexity of finding iso- and other morphisms for partial k-trees. *Discrete Mathematics*, 108: 343–364, 1992.
22. Haiko Müller. Hamiltonian circuits in chordal bipartite graphs. *Discrete Mathematics*, 156: 291–298, 1996.
23. Y. Okamoto, U. Takeaki, and R. Uehara, Counting the number of independent sets in chordal graphs. *Journal of Discrete Algorithms 6* (2008) 229–242.
24. N. Robertson and P.D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63: 65–110, 1995.

A The proof of Lemma 5

Lemma 5. *We can obtain in polynomial time a set $\mathcal{G}$ of graphs, such that G contains H as a contraction if and only if there exists a graph $G' \in \mathcal{G}$ that has an H-witness structure $\mathcal{W}'$ with $W'(y_j) = \{u_j(G')\}$ for $j = 1, \ldots, q$, where $u_1(G'), \ldots, u_q(G')$ are q specified mutually non-adjacent vertices of G' that are not cut-vertices of G'.*

Proof. We first explain how to create the set $\mathcal{G}$. We choose a pair (K_j, u_j) for each $y_j \in I_H$. This gives us an ordered set of q different pairs denoted by $S = \{(K_1, u_1), \ldots, (K_q, u_q)\}$. We determine all possible choices of sets S. If G contains H as a contraction, then G has an elementary witness structure $\mathcal{W}$ due to Lemma 4. Then one of our chosen sets will correspond to a set of q characteristic pairs for $\mathcal{W}$. We say that such a set S is a *right choice*. In order to determine this, we consider every set S one by one and work under the assumption that $\mathcal{W}$ exists. When considering a set S we modify G into a new graph G'. If in the end we have not discarded S then we put G' in $\mathcal{G}$.

First we check for a set S whether there exists a maximal clique K in G that contains two different vertices u_j and u_k. If so, then we discard S; otherwise should S be a right choice, then K contains a vertex $u_j \in W(y_j)$ and a vertex $u_k \in W(y_k)$, which is not possible due to Observation 2.

Suppose that we have not discarded S. For every pair (K_j, u_j) in S we do as follows. We first deal with the case in which u_j is a cut-vertex of G. In that case let T be the set of vertices of all components in $G - u_j$ except the vertices of the component that contains $K_j \setminus \{u_j\}$. Because $W(y_j)$ is not a cut-vertex of H, we find that $T \subseteq W(y_j)$. Hence, we can discard S if T contains some chosen vertex u_i. If this is the case, then T and u_j will be in the same witness bag should S be a right choice. Then, we can contract the vertices from T to u_j (or equivalently remove T from G). Hence, we may assume that u_j is not a cut-vertex of G.

If, after considering all pairs (K_j, u_j) in the way described as above, we have not discarded S then we keep it. Because some of the vertices u_j might have been cut-vertices, we might have modified G into a new graph G'. If we have not modified G, then we let $G' = G$. We put G' together with its associated set S in $\mathcal{G}$. Then, after considering all sets S we have obtained $\mathcal{G}$.

Because the graphs in $\mathcal{G}$ have been obtained from G by performing edge contractions and because we considered all possible cases, we find that G has an elementary H-witness structure $\mathcal{W}$ if and only if there exists a graph $G' \in \mathcal{G}$ that has a elementary H-witness structure $\mathcal{W}'$ such that its associated set $S(G') = \{(K_1, u_1), \ldots, (K_q, u_q)\}$ corresponds to a set of y_i-characteristic pairs for $\mathcal{W}'$.

Let $\mathcal{L}_{u_j}$ be the set of maximal cliques in $\mathcal{K}_{G'} \setminus \{K_j\}$ that contain u_j and that intersect with K_j. Because u_j is not a cut-vertex of G', every $K \in \mathcal{L}_{u_j}$ contains at least one other vertex of $K_j(G')$. This other vertex must be in $W'(C_H)$. Then, because $\mathcal{W}'$ is an elementary witness structure, all vertices in $K \setminus \{u_j\}$ must be in $W'(C_H)$. Hence, all neighbors of u_j are in $W'(C_H)$. Because $W'(y_j)$ is connected, this means that $W'(y_j) = \{u_j\}$, as desired.

The total number of different sets is $|V_G|^q$, which is polynomial because we assume q to be fixed. Also the processing time per set S is polynomial. Hence, constructing $\mathcal{G}$ costs polynomial time. We already showed that properties (i)-(iii) are valid for every $G' \in \mathcal{G}$. This completes the proof of Lemma 5. $\square$

B The Dynamic Programming of Section 3 in Details

Before we construct our dynamic programming algorithm, we first prove the following lemma. For doing this, we need the following notation. Let K be a node of $\mathcal{T}_{G^*}$. We denote by D_K the set of all descendants of K. For the parent K_p of K, we define D_{K_pK} as the set $D_K \cup \{K_p\}$.

Lemma 13. *If $\mathcal{W}$ is an H-witness structure for G with $W(y_j) = \{u_j\}$ for $j = 1, \ldots, q$, then the following two statements hold:*

(i) for each u_j, there exists a node K of $\mathcal{T}_{G^*}$ such that for any vertex $x_i \in C_H$ that is adjacent to y_j, there exists a vertex $v \in K \cap W(x_i)$ with $u_jv \in E_G$;

(ii) for each leaf node K of $\mathcal{T}_{G^*}$, there exists a vertex u_j such that K contains all neighbors of u_j in G.

Proof. We start by proving (i). First suppose that there exists a node K in $\mathcal{T}_{G^*}$ that contains all the neighbors of u_j in G. By definition, for every vertex $x_i \in C_H$ adjacent to y_j, the bag $W(x_i)$ contains a vertex v adjacent to u_j. Because K contains all neighbors of u_j, we find that K contains v, as desired. Now suppose that this is not the case.

Then there is a node K_1 such that $K_1 \setminus K_2$ and $K_2 \setminus K_1$ contain one or more neighbors of u_j for some node K_2. We choose a node K_1 with this property as close to the root of $\mathcal{T}_{G^*}$ as possible that, subject to this condition, contains maximum number of vertices adjacent to u_j. Consider a bag $W(x_i)$. Let $v \in W(x_i)$ be a vertex adjacent to u_j in G such that $v \notin K_1$. Then $v \in K_2 \setminus K_1$ for some node K_2. By the choice of K_1 and the chordality of G, either K_2 is a descendant of K_1, or K_2 is a descendant of the parent K of K_1.

Suppose that K_2 is a descendant of K_1. Let K' be the child of K_1 such that $K_2 \in D_{K_1K'}$. Because of the chordality of G, all vertices of $K_1 \cap K'$ are adjacent to u_j. By applying Lemma 7 inductively, we find that for all $v \in K_2 \setminus K_1$, there exists a vertex $t \in K_1 \cap K'$ with $W_t = W_v$. Hence, K_1 has a vertex $t \in K_1 \cap K'$ with the property that t is adjacent to u_j and $t \in W(x_i)$.

Assume now that K_2 is a descendant of the parent K of K_1, and denote by K' the child of K such that $K_2 \in D_{KK'}$. By chordality of G, we find that $K \cap K' = K \cap K_1$ and that all vertices of this set are adjacent to u_j. By Lemma 7, for all $v \in K_2 \setminus K$, there exists a vertex $t \in K \cap K' = K \cap K_1$ with $W_t = W_v$. Hence we conclude the same as before.

We now prove (ii). Let K be a leaf node of $\mathcal{T}_{G^*}$. Suppose that every u_j is adjacent to a vertex of $V_{G^*} \setminus K$. Then K contains a simplicial vertex of G. However, this is not possible as we deleted all such vertices. $\square$

We are now ready to construct our dynamic programming algorithm. Recall that the input contains the following:

- the graph G;
- a set of specified vertices $\{u_1, \dots, u_q\}$ of G;
- a clique tree $\mathcal{T}_{G^*}$ with a root K_r^*.

The algorithm returns `Yes` if G has an H-witness structure $\mathcal{W}$ with $W(y_j) = \{u_j\}$ for $j = 1, \dots, q$ that satisfies the property given in Lemma 7; in that case we say that G contains H as a *regular* contraction. The algorithm returns `No` otherwise. In order to simplify the description of the algorithm, we focus on the decision problem. However, our algorithm can be modified in order to get a desired H-witness structure.

Recall that y_j corresponds to the vertex $u_j \in V_G$. Let $d_j = d_H(y_j)$ for $j \in \{1, \dots, q\}$ and assume that the vertex y_j is adjacent to $x_{t_1^j}, \dots, x_{t_{d_j}^j}$ in H. Consider a node K of $\mathcal{T}_{G^*}$ with a parent K'. Let G_K be the subgraph of G^* induced by the set $\cup_{K^* \in D_K} K^*$ and $G_{K'K} = G^*[\cup_{K^* \in D_{K'K}} K^*]$ respectively. Let $K_1, \dots, K_s$ be the children of K. Let $G_i = G^*[\cup_{K^* \in D_{KK_i}} K^*]$ for $i \in \{1, \dots, s\}$ and $G_0 = G^*[K]$. For the node K, our algorithm consecutively constructs tables of data $R_i(K)$ for $i = 0, 1, \dots, s$ and then a table $R(K', K)$ (for the root node, only the tables $R_i(K_r^*)$ are constructed, and only $R_0(K)$ and $R(K', K)$ are constructed for leaves).

First, we describe the tables $R_i(K)$. The tables store vectors $(X_1, \dots, X_q)$ of the (possibly empty) ordered sets of vertices $X_j \subseteq K$ such that

(i) each X_j is either empty or X_j is an ordered set of vertices $\{x_{jt_1^j}, \dots, x_{jt_{d_j}^j}\} \subseteq K$;
(ii) for all pairs $x_{jt_l^j}, x_{j't_{l'}^{j'}}$, if $x_{t_l^j} \neq x_{t_{l'}^{j'}}$ then $x_{jt_l^j} \neq x_{j't_{l'}^{j'}}$;
(iii) for each u_j, if y_j is not adjacent to x_t then u_j is not adjacent to the vertices $x_{j't}$;

and there is a subgraph contraction of G_i to the subgraph G_0 with the following properties:

(iv) each vertex $v \in V_{G_i} \setminus V_{G_0}$ is contracted to a vertex of the set $\cup_{j=1}^q X_j$;
(v) for each vertex $x_{jt} \in X_j$, if x_{jt} is not adjacent to u_j in G then a vertex $v \in V_{G_i} \setminus V_{G_0}$ adjacent to u_j is contracted to x_{jt};
(vi) for each u_j, if y_j is not adjacent to x_t then vertices $v \in V_{G_i} \setminus V_{G_0}$ adjacent to u_j are not contracted to $x_{j't}$.

We use the following intuition here. If the set $X_j = \{x_{jt_1^j}, \dots, x_{jt_{d_j}^j}\} \neq \emptyset$ then the partial solution $(X_1, \dots, X_q)$ is supposed to be a part of a general solution (i.e. of a regular contraction of G to H) with the property that the vertices $x_{jt_1^j}, \dots, x_{jt_{d_j}^j}$ are included in the bags $W(x_{t_1^j}), \dots, W(x_{t_{d_j}^j})$.

In the similar way, the table $R(K', K)$ stores vectors $(\hat{X}_1, \dots, \hat{X}_q)$ of the (possibly empty) ordered sets of vertices $\hat{X}_j \subseteq K \cap K'$ such that

(i') each $\hat{X}_j$ is either empty or $\hat{X}_j$ is an ordered set of vertices $\{\hat{x}_{jt_1^j}, \dots, \hat{x}_{jt_{d_j}^j}\} \subseteq K$, but $\cup_{j=1}^q \hat{X}_j \neq \emptyset$;

(ii') for all pairs $\hat{x}_{jt_l^j}, \hat{x}_{j't_{l'}^j}$, if $x_{t_l^j} \neq x_{t_{l'}^{j'}}$ then $\hat{x}_{jt_l^j} \neq \hat{x}_{j't_{l'}^{j'}}$;

(iii') for each u_j, if y_j is not adjacent to x_t then u_j is not adjacent to the vertices $\hat{x}_{j't}$;

and there is a subgraph contraction of $G_{K'K}$ to the subgraph $G[K']$ with the following properties:

(iv') each vertex $v \in V_{G_K} \setminus K'$ is contracted to a vertex of the set $\cup_{j=1}^q \hat{X}_j$;

(v') for each vertex $\hat{x}_{jt} \in \hat{X}_j$, if $\hat{x}_{jt}$ is not adjacent to u_j in G then a vertex $v \in V_{G_K} \setminus K'$ adjacent to u_j is contracted to x_{jt};

(vi') for each u_j, if y_j is not adjacent to x_t then vertices $v \in V_{G_i} \setminus K'$ adjacent to u_j are not contracted to $\hat{x}_{j't}$.

Now we describe how the tables are created and maintained. We start with the tables $R_i(K)$.

Construction of $R_0(K)$. For each u_j, we consider all sets of vertices X_j such that either $X_j = \emptyset$ or $X_j = \{x_{jt_1^j}, \dots, x_{jt_{d_j}^j}\} \subseteq K$, and for each choice the vector $(X_1, \dots, X_q)$ is constructed. If it satisfies (i)–(iii) and, additionally, each chosen vertex $x_{jt} \in X_j$ is adjacent to u_j in G then the entry $(X_1, \dots, X_q)$ is included in $R_0(K)$.

Construction of $R_i(K)$ for $i > 0$. Assume now that the tables $R_{i-1}(K)$ and $R(K, K_i)$ are already constructed. For each pair of entries $(X_1, \dots, X_q) \in R_{i-1}(K)$ and $(\hat{X}_1, \dots, \hat{X}_q) \in R(K, K_i)$ such that at least one of sets $X_j, \hat{X}_j$ is empty for each $j \in \{1, \dots, q\}$, we construct the vector $(X_1 \cup \hat{X}_1, \dots, X_q \cup \hat{X}_q)$, and if this vector satisfies (i)—(iii) then it is included in $R_i(K)$.

Our next step is the construction of $R(K', K)$.

Construction of $R(K', K)$. Assume that the table $R_s(K)$ is already constructed. For each $(X_1, \dots, X_q) \in R_s(K)$, we consider all vectors $(\hat{X}_1, \dots, \hat{X}_q)$ with $\hat{X}_j \subseteq K \cap K'$ which satisfy (i')–(iii') such that

a) for each x_{jt}, if $x_{jt} \in K \cap K'$ then $x_{jt} = \hat{x}_{jt}$;
b) for all pairs $x_{jt}, x_{j't}$, if $x_{jt} = x_{j't}$ then $\hat{x}_{jt} = \hat{x}_{j't}$;
c) for each $v \in K \setminus (K' \cup (\cup_{j=1}^q X_j))$ and each set of vertices $\{u_{i_1}, \dots, u_{i_l}\}$ that v is adjacent to the vertices $u_{i_1}, \dots, u_{i_l}$, there is $\hat{x}_{jt}$ that x_t is adjacent to $y_{i_1}, \dots, y_{i_l}$;

and include them in $R(K', K)$.

Our algorithm constructs these tables starting from the leaves of $\mathcal{T}_{G^*}$. Finally, the table $R_s(K_r^*)$ is constructed for the root node K_r^* with s children. We analyze this table as follows. Let h be the number of vertices in C_H not adjacent with the vertices I_H. If $R_s(K_r^*)$ contains an entry $(X_1, \dots, X_q)$ with $X_j = \{x_{jt_1^j}, \dots, x_{jt_{d_j}^j}\}$ such that

a) for each $j \in \{1, \ldots, q\}$, $X_j \neq \emptyset$,
b) for each $v \in K_r^* \setminus (\cup_{j=1}^q X_j)$ and each set of vertices $\{u_{i_1}, \ldots, u_{i_l}\}$ that v is adjacent to the vertices $u_{i_1}, \ldots, u_{i_l}$, there is x_{jt} that x_t is adjacent to $y_{i_1}, \ldots, y_{i_l}$, and
c) there is at least h vertices in $K_r^* \setminus (\cup_{j=1}^q X_j)$ not adjacent to $u_1, \ldots, u_q$,

then the algorithm returns `Yes` answer, and it returns `No` otherwise.

Lemma 14. *The algorithm correctly decides in polynomial time whether G contains H as a regular contraction.*

Proof. Correctness of the algorithm follows from the description and Lemmas 7-13. Indeed, by Lemma 7, any regular contraction of G to H can be seen as a sequence of subgraph contractions of the graphs $G(K)$ to the induced subgraph $G[K]$ for the nodes K, performed starting from the leaves. It is straightforward to check that for any leaf K, the algorithm correctly constructs the tables $R_0(K)$ and $R(K', K)$ in such a way that they store all entries that satisfy conditions (i)–(iii) and (i')–(vi'), respectively. Then we apply inductive arguments. We use Lemma 8 to argue, that if we obtain simplicial vertices in the process then they can be contracted to the vertices of $G[K]$ arbitrary. Lemma 13 is used to prove that if G contains H as a regular contraction, then at some moment it is always possible to construct non-empty set X_j or $\hat{X}_j$, respectively.

To estimate the running time, observe that each of the tables $R_i(K)$ and $R(K', K)$ contains at most $2^q n^{pq}$ entries where $n = |V_G|$. It follows that the construction of each table can be done in time $n^{O(pq)}$. Since $\mathcal{T}_{G^*}$ has at most n nodes, the total running time is $n^{O(pq)}$.

C The Proof of Lemma 10

Lemma 10. *We can obtain in polynomial time a set $\mathcal{G}$ of graphs, such that G contains H as a contraction if and only if there exists a graph $G' \in \mathcal{G}$ that has a simple H-witness structure $\mathcal{W}'$ with as leaf bag vertices q specified vertices $u_1(G'), \ldots, u_r(G')$ that are of degree one in G' and that together form the set of all simplicial vertices of G'.*

Proof. We first explain how to create the set $\mathcal{G}$. We choose a vertex u_i for each z_i. This gives us a set $S = \{u_1, \ldots, u_q\}$ of q different vertices of G. We determine all possible choices of sets S. If G contains H as a contraction, then G has a simple witness structure $\mathcal{W}$ due to Lemma 9. Then one of our chosen sets will correspond to a set of q leaf bag vertices of $\mathcal{W}$. We say that such a set S is a *right choice*. In order to determine this, we consider every set S one by one and work under the assumption that $\mathcal{W}$ exists. When considering a set S we modify G into a new graph G'. If in the end we have not discarded S, then we put G' in $\mathcal{G}$.

First we check if a set $S = \{u_1, \ldots, u_q\}$ is independent. If not, then we discard S; otherwise, should S be a right choice, then two leaf bags of $\mathcal{W}$ are adjacent and this is not possible.

Suppose that S is an independent set. If S is a right choice, then all neighbors of each u_i are in the parent bag of $W(z_i)$. This means that we can contract an edge between any two adjacent neighbors of each u_i. We perform this operation as long as possible. We denote the resulting graph by G^*. Because G is chordal, G^* is chordal. We note that the neighbors of each u_i form an independent set in G^*; otherwise we could perform more edge contractions. Suppose that there exists a vertex u_i with more than one neighbor. Then, because G^* is chordal, u_i must be a cut-vertex of G^*. However, in that case, $W(z_i)$ must contain at least another vertex besides u_i, because $W(z_i)$ is a leaf bag. This is not possible, because $\mathcal{W}$ is a simple H-witness structure. Hence, we must discard S in this case.

Suppose that we have not discarded S, i.e., that every u_i has only one neighbor in G^*. Then every u_i is simplicial in G^*. Suppose that G^* contains a simplicial vertex $v \notin S$. We note that every neighbor of v is not in S, because the vertices in S have degree one in G^* and there are at least two vertices in S. Let w be a neighbor of v. We claim that we may contract the edge vw. This can be seen as follows. Let $W(x)$ denote the witness bag of $\mathcal{W}$ that contains v, and let $W(x')$ denote the witness bag of $\mathcal{W}$ that contains w. If $x = x'$, then contracting vw is allowed, as by definition of a witness structure we will contract $W(x)$ into a single vertex to obtain H.

Suppose that $x \neq x'$. We claim that in that case we can move v to $W(x')$ in order to contract vw. There are three properties of $\mathcal{W}$ to take into account. Firstly, $W(x)$ may become empty, i.e., when $W(x) = \{v\}$. Because $W(z_i) = \{u_i\}$ for $i = 1, \ldots, r$ already, $W(x)$ is not a leaf bag. Then the neighbors of v must go into at least two different witness bags. However, then H contains a triangle, because the set of neighbors of v forms a clique. This is not possible. Hence, $W(x)$ contains at least one more vertex. Secondly, $W(x)$ may become disconnected. However, because v is simplicial, v cannot be a cut-vertex of $W(x)$. Hence, $W(x)$ stays connected. For the same reason we not remove any witness edges either. Hence, we can also safely contract vw in this case.

We also contract an edge between any other simplicial vertex of G^* that is not in S and one if its neighbors, until we find that all simplicial vertices are in S. We denote the resulting graph by G'. After considering all sets S in this way we have obtained $\mathcal{G}$.

Because the graphs in $\mathcal{G}$ have been obtained from G by performing edge contractions and because we considered all possible cases, we find that G has a simple H-witness structure $\mathcal{W}$ if and only if there exists a graph $G \in \mathcal{G}$ that has a simple H-witness structure $\mathcal{W}'$ such that its associated set $S(G') = \{u_1, \ldots, u_r\}$ corresponds to the leaf bag vertices of $\mathcal{W}'$. We already showed that these vertices have degree one in G' and that G' has no other simplicial vertices.

The total number of different sets is $|V_G|^r$, which is polynomial, because r is fixed. Also the processing time per set S is polynomial. Hence, constructing $\mathcal{G}$ costs polynomial time. This completes the proof of Lemma 10. $\square$