Comparing universal covers in polynomial time

Jiří Fiala[1] and Daniël Paulusma[2]

[1] Charles University, Faculty of Mathematics and Physics,
DIMATIA and Institute for Theoretical Computer Science (ITI) ***,
Malostranské nám. 2/25, 118 00, Prague, Czech Republic.
`fiala@kam.mff.cuni.cz`
[2] Department of Computer Science, Durham University †,
Science Laboratories, South Road,
Durham DH1 3EY, England.
`daniel.paulusma@durham.ac.uk`

Abstract. The universal cover T_G of a connected graph G is the unique (possible infinite) tree covering G, i.e., that allows a locally bijective homomorphism from T_G to G. Universal covers have major applications in the area of distributed computing. It is well-known that if a graph G covers a graph H then their universal covers are isomorphic, and that the latter can be tested in polynomial time by checking if G and H share the same degree refinement matrix. We extend this result to locally injective and locally surjective homomorphisms by following a very different approach. Using linear programming techniques we design two polynomial time algorithms that check if there exists a locally injective or a locally surjective homomorphism, respectively, from a universal cover T_G to a universal cover T_H. This way we obtain two heuristics for testing the corresponding locally constrained graph homomorphisms. As a consequence, we have obtained a new polynomial time algorithm for testing (subgraph) isomorphism between universal covers, and for checking if there exists a role assignment (locally surjective homomorphism) from a given tree to an arbitrary fixed graph H.

*** Supported by the Ministry of Education of the Czech Republic as project 1M0021620808.
† Supported by EPSRC as project EP/D053633/1.

1 Introduction

In this paper, we consider simple, undirected, possibly infinite but connected graphs. See [5] for undefined graph terminology. A *(graph) homomorphism* $f : G \to H$ from a graph $G = (V_G, E_G)$ to a graph $H = (V_H, E_H)$ is a mapping $V_G \to V_H$ such that $(f(u), f(v)) \in E_H$ whenever $(u, v) \in V_G$. Graph homomorphisms have a great deal of applications in graph theory, computer science and other fields, see the monograph [16].

A graph homomorphism f from a graph G to a graph H can be required to satisfy some local constraint [9]. If, for every $u \in V_G$ the restriction of f, i.e. the mapping $f_u : N(u) \to N(f(u))$, is bijective, we say that f is *locally bijective* [1, 18], and we write $G \xrightarrow{B} H$. If, for every $u \in V_G$, f_u is injective, we say that f is *locally injective* [10, 11], and we write $G \xrightarrow{I} H$. If, for every $u \in V_G$, f_u is surjective, we say that f is *locally surjective* [13, 19], and we write $G \xrightarrow{S} H$.

Locally bijective homomorphisms, also called graph coverings, originally arose in topological graph theory [21], and have applications in distributed computing [4], in recognizing graphs by networks of processors [2], and in constructing highly transitive regular graphs [3]. Locally injective homomorphisms, also called partial graph coverings, have been studied due to their applications in models of telecommunication [11], in distance constrained labelings of graphs with applications to frequency assignment [12], and as indicators of the existence of homomorphisms of derivate graphs (line graphs) [23]. Locally surjective homomorphisms, also called role assignments, have applications in distributed computing [6] and social science [8, 25].

The main computational question is whether for every graph H the problem of deciding if an input graph G has a homomorphism of given type $* = B, I$ or S to the fixed graph H can be classified as either NP-complete or polynomially solvable. For the locally surjective homomorphisms this classification is known [13], with the problem for every connected H on at least three vertices being NP-complete. For the locally bijective and injective cases there are many partial results, see e.g. [11, 18], but even conjecturing a classification for these two cases is problematic. In this paper, we continue the study started in [14] in order to get more insight in the structure of these computational issues.

1.1 Problem formulation

The existence of a locally constrained homomorphism imposes a partial order on the class of connected graphs $\mathcal{C}$ for each of the three local constraints B, I, and S [14]. We can relax these three orders in two different ways. This leads to two different heuristics for testing if $G \xrightarrow{*} H$ for two given graphs G and H under each type $* = B, I, S$.

Firstly, we can transform the partial orders from the domain of finite graphs to the domain of matrices. An *equitable partition* of a connected graph G is a partition of its vertex set in blocks $B_1, \dots, B_k$ such that each vertex in each B_i has the same number $m_{i,j}$ of neighbors in B_j, and we call the $k \times k$ matrix $M = (m_{i,j})_{1 \leq i,j \leq k}$ a *degree matrix* of G. We say that a vertex u is of the *i-th sort* if $u \in B_i$. Equitable partitions are well-known in algebraic graph theory, see e.g. [15]. Note that the degree refinement matrix of G, is the degree matrix corresponding to the coarsest equitable partition of G (in which the blocks are ordered in a unique way), and an adjacency matrix of G can be seen as a degree matrix with the maximum number of rows.

Let $\mathcal{M}$ be the set of all degree matrices. We define three relations $(\mathcal{M}, \xrightarrow{\exists B})$, $(\mathcal{M}, \xrightarrow{\exists I})$ and $(\mathcal{M}, \xrightarrow{\exists S})$ imposed on the set of degree matrices by the existence of graph homomorphisms of the corresponding local constraint, i.e., $M \xrightarrow{\exists *} N$ if and only if $G \xrightarrow{*} H$ for two graphs $G, H \in \mathcal{C}$ that have degree matrix M and N, respectively. All three relations are partial orders [14], and a successful matrix comparison of each type is a necessary condition for the corresponding graph comparison.

Secondly, we can transform the partial orders from the domain of finite graphs to the domain of possibly infinite trees. The *universal cover* T_G of a connected graph G is the only tree that allows a locally bijective homomorphism $T_G \xrightarrow{B} G$. A generic construction of the universal cover takes as vertices of T_G all finite walks in G that start from an arbitrary fixed vertex in G and that does not traverse the same edge in two consecutive steps. Two such vertices are adjacent in T_G if the associated walks differ only in the presence of the last edge. The required homomorphism $T_G \xrightarrow{B} G$ can be taken as the mapping that assigns every walk its last vertex. One can easily see that the universal cover is unique upto an isomorphism (in particular, if we take walks that start in another fixed vertex). As a matter of fact, if two subtrees of a universal cover rooted at two different vertices are isomorphic to depth $n-1$, then they are isomorphic to all depths [24]. Universal covers are also called infinite unfoldings or views of graphs and have applications in finite automata theory[22], distributed computing [17, 26] and existential pebble games [7].

Also universal covers can be equipped with a structure that impose a necessary condition for the existence of a locally constrained homomorphism. There are two options: either the existence of a locally constrained homomorphism or a simple inclusion (as a subtree). In the latter case, $T_G = T_H$, $T_G \subseteq T_H$, and $T_G \supseteq T_H$ are necessary conditions for $G \xrightarrow{B} H$, $G \xrightarrow{I} H$ and $G \xrightarrow{S} H$, respectively, see [14] for more details.

Moreover, a result in [14] states that the universal cover T_G can be constructed from any degree matrix M of G in the following way. We take as root a vertex corresponding to row 1 of M, thus of the 1st sort, and inductively adding a new level of vertices while maintaining the property that each vertex of the i-th sort has exactly $m_{i,j}$ neighbors of the j-th sort. Hence, a successful universal cover comparison is a necessary condition for the corresponding graph comparison as well. More precisely, we have shown the forward implications in the following theorem.

Theorem 1. *Let G and H be connected graphs with degree matrices M and N, resp. Then the following holds:*

$$\begin{array}{c}
G \xrightarrow{B} H \Longrightarrow M \xrightarrow{\exists B} N \Longleftrightarrow T_G \xrightarrow{B} T_H \Longleftrightarrow T_G = T_H \\
G \xrightarrow{I} H \Longrightarrow M \xrightarrow{\exists I} N \Longrightarrow T_G \xrightarrow{I} T_H \Longleftrightarrow T_G \subseteq T_H \\
G \xrightarrow{S} H \Longrightarrow M \xrightarrow{\exists S} N \Longrightarrow T_G \xrightarrow{S} T_H \Longrightarrow T_G \supseteq T_H
\end{array}$$

The backward implications in Theorem 1 for locally bijective homomorphism are consequences of the theorem of Leighton [20]. The equivalence $T_G \xrightarrow{I} T_H \Longleftrightarrow T_G \subseteq T_H$ follows form the fact that a locally injective homomorphisms between two trees is indeed globally injective [14].

Observe that $C_4 \not\xrightarrow{*} C_3$ while both graphs allow the 1×1 degree matrix $M = (2)$. This example excludes the implication $G \xrightarrow{*} H \Longleftarrow M \xrightarrow{\exists *} N$ for $* = B, I, S$. If G itself is a tree then $T_G = G$. We then find that $T_G \xrightarrow{S} T_H \not\Longleftarrow T_G \supseteq T_H$ for the choice $G = P_4$, $H = P_3$, since $P_4 \supseteq P_3$ but $P_4 \not\xrightarrow{S} P_3$. This example shows that the relations $T_G \xrightarrow{S} T_H$ and $T_G \supseteq T_H$ are different. By using linear programming techniques, the backward implication $M \xrightarrow{\exists I} N \Longleftarrow T_M \xrightarrow{I} T_N$ can be excluded [14]. So, the inclusion of universal covers does not imply the relation on matrices for the locally injective constraint. What about the remaining backward implication?

Question 1. Does there exist a counter example for the backward implication $M \xrightarrow{\exists S} N \Longleftarrow T_G \xrightarrow{S} T_H$ in Theorem 1?

The problem of deciding $G \xrightarrow{*} H$ is NP-complete for all three local constraints, and remains NP-hard for many particular fixed targets H, as we mentioned earlier on. We have shown that $M \xrightarrow{\exists B} N$ can be verified in polynomial time, but so far only membership to the class NP could be shown for the matrix comparison problem $M \xrightarrow{\exists *} N$ for $* = I, S$ [14]. It is not expected that a polynomial algorithm would solve these two problems. Testing if $T_G = T_H$ can be done in polynomial time by checking if G and H share the same the degree refinement matrix [2]. Especially given the above, it would be useful to have a polynomial heuristic for checking the other universal problem comparisons as well.

Question 2. How hard is it to decide if $T_G \xrightarrow{I} T_H$ (or equivalently $T_G \subseteq T_H$) holds and to decide if $T_G \xrightarrow{S} T_H$ holds for two given connected graphs G and H?

In this paper we answer Question 1 in Section 2 as well as Question 2 in Section 3.

2 Excluding the remaining implication

We show that the relation $T_M \xrightarrow{S} T_N$ lies strictly between $M \xrightarrow{\exists S} N$ and $T_M \supseteq T_N$.

Proposition 1. *For degree matrices*

$$M = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \text{and} \qquad N = \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}$$

it holds that $T_M \xrightarrow{S} T_N$ but $M \not\xrightarrow{\exists S} N$.

Proof. Observe first that N is a matrix of a finite tree T_N and no other connected simple graph allows this degree matrix. The infinite tree T_M consist of pairwise disjoint paths that are of infinite length and induced by vertices of the first sort (white vertices). These paths are linked by vertices of the second sort (each is adjacent to three paths) and every vertex of the second sort is joined to the middle vertex of a unique P_3. The trees T_M and T_N together with a homomorphism witnessing $T_M \xrightarrow{S} T_N$ are depicted in Fig. 1.

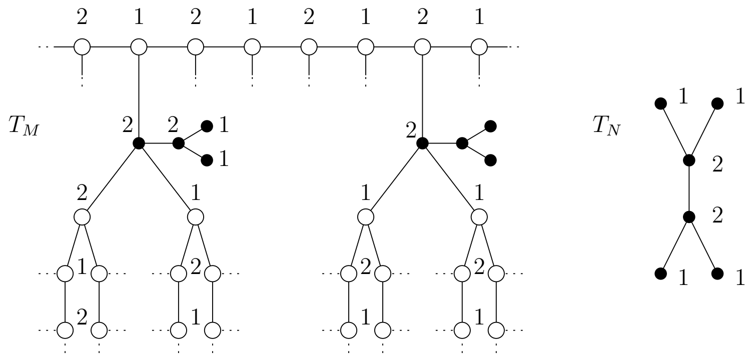

Fig. 1. Showing $T_M \xrightarrow{S} T_N$. White vertices in T_M are of the 1st sort.

This homomorphism is obtained inductively. We first map one infinite white path into T_N such that the sorts of the images alternate. Every vertex u of the second sort in T_M must be mapped on a vertex of the second sort in T_N so that the homomorphism can be extended to the pending claw.

Then, depending of whether the image of the already processed neighbor of u was of the first or of the second sort, we extend the mapping to the two infinite white paths that contains the remaining two neighbors of u. Both cases are depicted in Fig. 1.

Now, in order to obtain a contradiction, assume that a finite graph G with degree matrix M and a mapping $f : G \xrightarrow{S} T_N$ exists (recall that the target graph T_N is unique for this choice of N). Consider the vertices of the first sort of G, call them red. These red vertices induce a disjoint union of cycles in G.

Denote by a the number of red vertices u such that $f(u)$ is of the first sort in T_N and call them light-red. Analogously, let b be the number of red vertices v such that $f(v)$ is of the second sort, and call them dark-red. Since N prescribes that both red neighbors of every light-red u must be dark we have $a \leq b$.

On the other hand, due to the pending claws (which also exist in G), every vertex of the third sort in G is mapped to a vertex of the second sort in T_N, and every vertex of the fourth sort in G is mapped to a vertex of the first sort in T_N. Then every vertex u' of the second sort in G is mapped to a vertex of the second sort in T_N. Since already its neighbor of the third sort is mapped to a vertex of the second sort in T_N, u' must have at least two light-red neighbors and, consequently, at most one dark-red neighbor. Hence, $a \geq 2b$ which is in contradiction with $a \leq b$. We conclude that $M \overset{S}{\not\longrightarrow} N$. □

3 Testing locally injective and surjective homomorphisms between universal covers

In this section we focus on the decision problems whether $T_M \xrightarrow{*} T_N$ holds for local constraints $* = I, S$. As the algorithms are almost the same for both constraints, we treat both cases simultaneously, pointing only at the differences where the particular local constraint plays different role. We first need some new terminology. For an integer $k \geq 1$ we define $[k] := \{1, 2, \dots, k\}$ and abbreviate $[k] \times [l]$ by $[k \times l]$.

Definition 1. *Let M and N be two degree matrices of order k and l, resp. We say that a vector $\mathbf{p}^{r,s}$ consisting of kl nonnegative integers is a* distribution row *for indices $(r, s) \in [k \times l]$ if the condition 1 holds. A distribution row $\mathbf{p}^{r,s}$ is called* injective *if in addition condition 2 holds. It is called* surjective *if in addition conditions 3 and 4 hold.*

$$\sum_{j=1}^{l} p_{i,j}^{r,s} = m_{r,i} \qquad \text{for all } i \in [k], \tag{1}$$

$$\sum_{i=1}^{k} p_{i,j}^{r,s} \leq n_{s,j} \qquad \text{for all } j \in [l], \tag{2}$$

$$n_{s,j} \geq 1 \quad \Longrightarrow \quad \sum_{i=1}^{k} p_{i,j}^{r,s} \geq n_{s,j} \qquad \text{for all } j \in [l], \tag{3}$$

$$n_{s,j} = 0 \quad \Longrightarrow \quad \sum_{i=1}^{k} p_{i,j}^{r,s} = 0 \qquad \text{for all } j \in [l]. \tag{4}$$

As an example, consider the matrices M and N from Proposition 1. The locally surjective homomorphism from T_M and T_N depicted in Figure 1 defines exactly the following surjective distribution rows:

$$\begin{array}{l} \mathbf{p}^{1,1} = (0,2,0,1,0,0,0,0) \\ \mathbf{p}^{1,2} = (2,0,0,1,0,0,0,0) \\ \mathbf{p}^{2,2} = (2,1,0,0,0,1,0,0) \\ \mathbf{p}^{3,2} = (0,0,0,1,0,0,2,0) \\ \mathbf{p}^{4,1} = (0,0,0,0,0,1,0,0) \end{array}$$

Distribution rows play a central role in the NP algorithms for the degree matrix comparison problems $M \xrightarrow{\exists I} N$ and $M \xrightarrow{S} N$ [14]. Suppose $G \xrightarrow{*} H$ via f is indeed a witness for $M \xrightarrow{\exists *} N$ for $* \in \{I, S\}$. Let f map $u \in V_G$ of the i-th sort to $v \in V_H$ of the j-th sort, and denote the number of neighbors of the i-th sort in $N_G(u)$ that are mapped to neighbors of the j-th sort in $N_H(v)$ by $p^{r,s}_{i,j}$. Then the vector $\mathbf{p}^{r,s}$ defined by entries $p^{r,s}_{i,j}$ is a (surjective or injective) distribution row that we call suitable. Our NP algorithms try to identify suitable distribution rows. The difficulty is that there may be exponentially many distribution rows. Therefore, these algorithms could only use the nondeterministic choice of suitable distribution rows to verify whether $M \xrightarrow{\exists *} N$ holds for $* = I, S$, respectively, see [14] for more details. However, for the decision problem on the existence of a locally constrained homomorphism between universal covers we prove that we may reduce the number of suitable distribution rows to only a polynomial number. For showing this we need some more terminology. For a degree matrix M we say that matrix rows r and i are *adjacent* if $m_{r,i} > 0$.

Definition 2. *We say that a distribution row* $\mathbf{p}^{r,s}$ *is* a witness of type (s, j) *for (adjacent) matrix rows r and i if* $\mathbf{p}^{r,s}_{i,j} \geq 1$.

Definition 3. *We say that a distribution row* $\mathbf{p}^{r,s}$ respects the allowed set $X \subseteq [k \times l]$ *if* $\mathbf{p}^{r,s}_{i',j'} \geq 1$ *implies* $(i', j') \in X$ *for all* $(i', j') \in [k \times l]$.

Note that if $\mathbf{p}^{r,s}$ is a witness of type (s, j) for matrix rows r and i that respects an allowed set X then $(i, j) \in X$. We need the following lemma for our algorithms.

Lemma 1. *For given r and i the existence of an injective or surjective witness $p^{r,s}$ of type (s, j) respecting an allowed set X can be tested in a polynomial time.*

Proof. We can do this by translating the problem to the integer flow problem. It is well-known that this problem can be solved in polynomial time on flow networks with integer edge capacities (if such a network has a flow, then this flow may be assumed to be integer). We first define our auxiliary flow network F and then explain it afterwards. We let $V_F = \{p, u_{i'}, v_{j'}, q \mid (i', j') \in [k \times l]\}$ and $E_F = \{(p, u_{i'}), (u_{i'}, v_{j'}), (v_{j'}, q) \mid (i', j') \in [k \times l]\}$. The sought flow g goes from p to q and must satisfy the following edge constraints:

$$g(p, u_{i'}) = m_{r,i'} \qquad \text{for } * = I: \; g(v_{j'}, q) \leq n_{s,j'}$$

$$g(u_{i'}, v_{j'}) \begin{cases} \geq 1 & \text{if } (i', j') = (i, j) \\ = 0 & \text{if } (i', j') \notin X \\ \geq 0 & \text{otherwise} \end{cases} \qquad \text{for } * = S: \; g(v_{j'}, q) \begin{cases} \geq n_{s,j'} & \text{if } n_{s,j'} \geq 1 \\ = 0 & \text{if } n_{s,j'} = 0 \end{cases}$$

We claim that F has an integer flow g if and only if there exists an injective, or respectively, surjective witness $p^{r,s}$ of type (s, j) for r and i respecting X. First suppose F allows an integer flow g. Choose $\mathbf{p}^{r,s} = g$. Because $\sum_{j'=1}^{l} p^{r,s}_{i',j'} = \sum_{j'=1}^{l} g(u_{i'}, v_{j'}) = g(p, u_{i'}) = m_{r,i'}$ for all $i' \in [k]$, $\mathbf{p}^{r,s}$ is a distribution row. For all $j' \in [l]$, $\sum_{i'=1}^{k} p^{r,s}_{i',j'} = \sum_{i'=1}^{k} g(u_{i'}, v_{j'}) = g(v'_j, q)$, which is at most $n_{s,j'}$ if $* = I$, at least $n_{s,j'}$ if $* = S$ and $n_{s,j'} \geq 1$, and 0 otherwise, $p^{r,s}$ is injective or surjective, respectively. Since $p^{r,s}_{i,j} = g(u_i, v_j) \geq 1$, $\mathbf{p}^{r,s}$ is a witness. Finally, since $p^{r,s}_{i',j'} = g(u_{i'}, v_{j'}) = 0$ for all $(i', j') \notin X$, $\mathbf{p}^{r,s}$ respects X.

Now suppose there exists an injective, or respectively, surjective witness $p^{r,s}$ of type (s, j) for r and i respecting X. By Definition 2, $p^{r,s}_{i,j} \geq 1$. It is easy to verify that $p^{r,s}$ satisfies the other edge constraints in F as well. Hence F allows $p^{r,s}$ as integer flow. □

Our two algorithms can now be presented as one generic iterative algorithm.

Algorithm 1: The test whether $T_M \xrightarrow{*} T_N$ holds for $* = I$ or S

Input: Degree matrices M and N
Parameter: Local constraint $* \in \{I, S\}$
initialize $X^{r,s} = \{(i,j) \mid m_{r,i} > 0 \text{ and } n_{s,j} > 0\}$ for all $(r,s) \in [k \times l]$;
repeat
  foreach $(r,s) \in [k \times l]$ *and* $(i,j) \in X^{r,s}$ **do**
    if (r,i) *has no witness of type* (s,j) *respecting* $X^{r,s}$ **then**
      remove (i,j) from $X^{r,s}$ and remove (r,s) from $X^{i,j}$;
    end
  end
until *no removal happens during the whole* **foreach** *loop* ;
if *there exists an* $r \in [k]$ *with* $X^{r,s}$ *empty for all* $s \in [l]$ **then**
  return $T_M \stackrel{*}{\not\to} T_N$
else
  return $T_M \xrightarrow{*} T_N$
end

Theorem 2. *Algorithm 1 is correct and runs in polynomial time.*

Proof. For each $X^{r,s}$, one iteration of Algorithm 1 takes polynomial time due to Lemma 1. Since the number of different allowed sets $X^{r,s}$ is kl, a complete iteration, i.e., an iteration over all $X^{r,s}$, then takes polynomial time as well. At the start of the algorithm each $X^{r,s}$ contains at most kl elements, and after each complete iteration the size of each $X^{r,s}$ has never increased. Since the algorithm finishes as soon as all $X^{r,s}$ have stable size, the number of iterations is at most kl. We conclude that Algorithm 1 runs in polynomial time.

We now show that Algorithm 1 is correct. Suppose $T_M \xrightarrow{*} T_N$ via f. Then f induces witnesses of type (s,j) for all adjacent matrix rows r, i such that (r,i) has a witness of type (s,j) if and only if (i,r) has a witness of type (j,s). Hence f defines nonempty sets $X^{r,s}$ for all matrix rows r.

It remains to show that if Algorithm 1 terminates in the affirmative state, then a locally constrained homomorphism $f : T_M \xrightarrow{*} T_N$ can be constructed. Pick an arbitrary vertex $u \in T_M$. Let u be of the r-th sort. By definition of the algorithm, there exists a (final) allowed set $X^{r,s} \neq \emptyset$. Define $f(u) = v$ for any $v \in T_N$ that is of the s-th sort. Choose an arbitrary $(i,j) \in X^{r,s}$. By definition of $X^{r,s}$, we can find a witness $\mathbf{p}^{r,s}$ for (r,i) of type (s,j) respecting $X^{r,s}$.

We use $\mathbf{p}^{r,s}$ to extend f. By definition of $\mathbf{p}^{r,s}$, for every $p^{r,s}_{i',j'} \geq 1$, we can let f map $p^{r,s}_{i',j'}$ different neighbors of u that all are of the i'-th sort onto neighbors of v that all are of the j'-the sort in such a way that, from $N(u)$ to $N(v)$, f is injective when $* = I$, and surjective when $* = S$. Whenever the mapping f is defined along an edge (u, u'), we iteratively extend f to the whole neighborhood $N(u')$ of u' by the same procedure as above in case $N(u') \supsetneq \{u\}$. We only have to make sure to choose a witness $p^{i',j'}$ for (i', r) of type (j', s), where $u, u', f(u)$ and $f(u')$ are of the r, i', s- and j'-th sort respectively. Then $p^{i',j'}_{r,s} \geq 1$ by definition of a witness, and indeed we can use $\mathbf{p}^{i',j'}$ to extend our mapping f that already maps $u \in N(u')$ of the r-th sort to $v \in N(f(u'))$ of the s-th sort. The reason why such a witness $\mathbf{p}^{i',j'}$ exists follows from the reciprocal removal of pairs (i', j') from $X^{r,s}$ and (r, s) from $X^{i',j'}$. When the condition of the **repeat** loop is satisfied, it holds that

$$
\begin{gathered}
(r, i') \textit{ has a witness of type } (s, j') \textit{ respecting } X^{r,s} \\
\text{if and only if} \\
(i', r) \textit{ has a witness of type } (j', s) \textit{ respecting } X^{i',j'}.
\end{gathered}
$$

□

We are even able to construct in polynomial time a locally constrained homomorphism $f : T_M \xrightarrow{*} T_N$ if Algorithm 1 approves that $T_M \xrightarrow{*} T_N$. This can be seen as follows. We use the method described in the proof of Theorem 2 to construct f. Finding witnesses respecting certain allowed sets can be done in polynomial time using the flow network of the proof of Lemma 1. If f is defined along edge (u, u') then we always choose for the same extension of f on $N(u')$, i.e., how we extend f only depends on the sort of u and the sort of u'. As it is sufficient to keep only at most kl possibilities, the claim follows.

4 Conclusions

We have answered questions 1 and 2 of Section 1.1 in Proposition 1 and Theorem 2, respectively. We conclude with some other applications.

The H-Role Assignment problem asks whether $G \xrightarrow{S} H$ for a graph G and a fixed target graph H. This problem is NP-complete for all connected graphs H on at least three vertices [13]. It becomes polynomially solvable for every fixed target H when restricted to the class of trees. This follows from Theorem 2, and the fact that $T \stackrel{S}{\not\to} G$ if T is a tree and G contains a cycle, together with the fact that $T_G = T$ for every tree T. Since $T_G \xrightarrow{I} T_H$ if and only if $T_G \subseteq T_H$, Algorithm 1 tests for infinite subtree isomorphism as well. Since $T_G = T$ for every tree T, it can also be used for (sub-)tree isomorphism for finite trees, especially if these trees can be encoded in terms of degree (refinement) matrices independent of their original size (as otherwise much faster algorithms exist). Finally, we note that there exist matrices that are not the degree matrix of a finite graph. If such a matrix M has the property that $m_{i,j} > 0$ whenever $m_{j,i} > 0$ then it still possible to construct a universal cover T_M of M (or disjoint submatrices of M) in the same way as before. Algorithm 1 can then be used for universal cover comparison of those matrices as well.

Acknowledgments. We thank Jan Arne Telle for fruitful discussions on this topic.

References

1. Abello, J., Fellows, M. R., and Stillwell, J. C. On the complexity and combinatorics of covering finite complexes. *Australian Journal of Combinatorics 4* (1991), 103–112.
2. Angluin, D. Local and global properties in networks of processors. In *Proceedings of the 12th ACM Symposium on Theory of Computing* (1980), 82–93.
3. Biggs, N. Constructing 5-arc transitive cubic graphs. *Journal of London Mathematical Society II. 26* (1982), 193–200.
4. Bodlaender, H. L. The classification of coverings of processor networks. *Journal of Parallel Distributed Computing 6* (1989), 166–182.
5. Bondy, J.A., and Murty, U.S.R. *Graph Theory with Applications.* Macmillan, London and Elsevier, New York, 1976.
6. Chalopin, J., Métivier, Y., and Zielonka W., Local computations in graphs: the case of cellular edge local computations. *Fundamenta Informaticae 74* (2006), 85–114.
7. Dantchev, S., Martin, B.D., and Stewart, I.A., On non-definability of unsatisfiability, manuscript.
8. Everett, M. G., and Borgatti, S. Role coloring a graph. *Mathematical Social Sciences 21* (1991), 183–188.
9. Fiala, J., Heggernes, P., Kristiansen, P., and Telle, J. A. Generalized H-coloring and H-covering of trees. *Nordic Journal of Computing 10* (2003), 206–224
10. Fiala, J., and Kratochvíl, J. Complexity of partial covers of graphs. In *Algorithms and Computation, 12th ISAAC '01*, LNCS 2223, 537–549.
11. Fiala, J., and Kratochvíl, J. Partial covers of graphs. *Discussiones Mathematicae Graph Theory 22* (2002), 89–99.

12. Fiala, J., Kratochvíl, J., and Kloks, T. Fixed-parameter complexity of λ-labelings. *Discrete Applied Mathematics 113* (2001), 59–72.
13. Fiala, J., and Paulusma, D. A complete complexity classification of the role assignment problem. *Theoretical Computer Science 349* (2005), 67-81.
14. Fiala, J., Paulusma, D., and Telle, J.A. Locally constrained graph homomorphisms and equitable partitions, to appear in *European Journal of Combinatorics.* see http://www.dur.ac.uk/daniel.paulusma/Papers/Submitted/locallycon.pdf
15. Godsil, C. *Algebraic Combinatorics.* Chapman and Hall, 1993.
16. Hell, P., and Nešetřil, J. *Graphs and Homomorphisms.* Oxford University Press, 2004.
17. Kranakis, E., Krizanc, D., and Van den Berg, J. Computing boolean functions on anonymous networks. *Information and Computation 114* (1994), 214–236.
18. Kratochvíl, J., Proskurowski, A., and Telle, J. A. Covering regular graphs. *Journal of Combinatorial Theory B 71* (1997), 1–16.
19. Kristiansen, P., and Telle, J. A. Generalized H-coloring of graphs. In *Algorithms and Computation, 11th ISAAC '01*, LNCS 1969, 456–466.
20. Leighton, F. T. Finite common coverings of graphs. *Journal of Combinatorial Theory B 33* (1982), 231–238.
21. Massey, W. S. *Algebraic Topology: An Introduction.* Harcourt, 1967.
22. Moore, E.F. Gedanken-experiments on sequential machines. *Ann. Math. Studies 34* (1956), 129–153.
23. Nešetřil, J. Homomorphisms of derivative graphs. *Discrete Mathematics 1* (1971), 257–268.
24. Norris, N. Universal covers of graphs: isomorphism to depth $n-1$ implies isomorphism to all depths. *Discrete Applied Mathematics 56* (1995), 61–74.
25. Roberts, F. S., and Sheng, L. How hard is it to determine if a graph has a 2-role assignment? *Networks 37*, (2001), 67–73.
26. Yamashita, M., and Kameda, T., Computing on anonymous networks: Part I - Characterizing the solvable cases. *IEEE Transactions on parallel and distributed systems 7* (1996), 69–89.