Polynomial Time Inductive Inference of Cograph Pattern Languages from Positive Data

Yuta Yoshimura[1], Takayoshi Shoudai[1], Yusuke Suzuki[2], Tomoyuki Uchida[2], and Tetsuhiro Miyahara[2]

[1] Department of Informatics, Kyushu University, Japan
{yuuta.yoshimura,shoudai}@inf.kyushu-u.ac.jp
[2] Department of Intelligent Systems, Hiroshima City University, Japan
{y-suzuki@,uchida@,miyares11@ml.info.}hiroshima-cu.ac.jp

Abstract. A cograph is a graph which can be generated by disjoint union and complement operations on graphs, starting with a single vertex graph. Toward effective data mining for graph structured data, we introduce a graph pattern expression, called a *cograph pattern*, based on cographs. We show that the class of cograph pattern languages is polynomial time inductively inferable from positive data.

1 Introduction

A *cograph* (complement reducible graph) is a graph which can be generated by disjoint union and complement operations on graphs, starting with a single vertex graph. Research results on cographs include recognition algorithm for cographs [3] and properties of cographs [2]. Any graph can become a cograph by adding edges. Some results on a method for adding a minimal number of such edges are obtained. Since a cograph has many useful properties, it is known that several problems which are intractable for general graphs, such as graph isomorphism problem, graph coloring problem and Hamiltonian cycle problem, are solvable in polynomial time for cographs.

In this paper, we introduce a *cograph pattern* which is an expression for common structures in graph databases. A cograph pattern is a graph pattern having structured variables which can be substituted by arbitrary cographs. For a cograph pattern g, the *cograph pattern language* of g is the set of all cographs obtained from g by substituting arbitrary cographs for all variables in g. A cograph pattern has a unique representation of a rooted tree. This representation is called a *cotree pattern*. We give examples of cograph patterns and their corresponding cotree patterns in Fig. 1. Firstly, we propose a polynomial time pattern matching algorithm for cograph patterns, based on cotree patterns, a polynomial time isomorphism algorithm for cographs [2] and a polynomial time pattern matching algorithm for linear interval graph patterns [7]. Secondly, we give a polynomial time algorithm for the minimal language problem which is, given a set S of cographs, to find a cograph pattern g such that the language of g is minimal among all cograph pattern languages which contain all cographs in

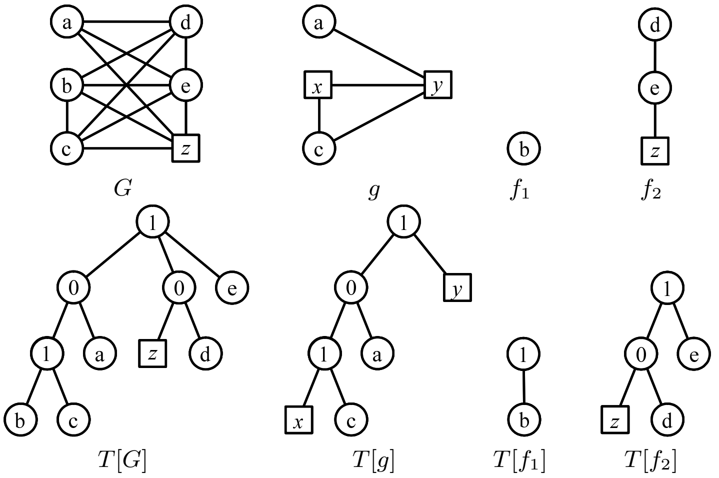

Fig. 1. G, g, f_1, and f_2 are cograph patterns which have cotree patterns $T[G]$, $T[g]$, $T[f_1]$, and $T[f_2]$, respectively. We use square boxes to describe variables of cograph patterns and cotree patterns.

S. Finally, we show that the class of cograph pattern languages is polynomial time inductively inferable from positive data.

2 Preliminaries

For a graph G, the vertex and edge sets of G are denoted by $V(G)$ and $E(G)$, respectively. For a subset U of $V(G)$, an *induced subgraph* of G w.r.t. U, denoted by $G[U]$, is the subgraph F of G such that $V(F) = U$ and $E(F) = \{\{u,v\} \in E(G) \mid u,v \in U\}$. For two graphs G_1 and G_2, a *union graph* of G_1 and G_2, denoted by $G_1 \cup G_2$, is the graph having the vertex set $V(G_1) \cup V(G_2)$ and the edge set $E(G_1) \cup E(G_2)$. For a graph G having no cycle and self-loop, a *complement graph* of G, denoted by $\bar{G}$, is the graph having the vertex set $V(G)$ and the edge set $\{\{u,v\} \mid u,v,\in V(G), \{u,v\} \notin E(G)\}$.

Definition 1. (**Cograph pattern**) Let Σ and $\mathcal{X}$ be alphabets. An element of $\mathcal{X}$ is called a *variable label* and a vertex labeled with a variable label is called a *variable*. Then, a *complement reducible graph pattern* (*cograph pattern*, for short) is a vertex-labeled undirected graph over $\Sigma \cup \mathcal{X}$ recursively defined as follows.

1. A single vertex labeled with an element in $\Sigma \cup \mathcal{X}$ is a cograph.
2. The union graph $G_1 \cup G_2 \cup \cdots \cup G_k$ of k cographs $G_1, G_2, \ldots, G_k$ is a cograph (*Disjoint Union Operation*).
3. The complement graph of a cograph is a cograph (*Complement Operation*).

A cograph pattern g is denoted by a triplet $(V(g), E(g), H(g))$ consisting of a set $V(g)$ of vertices labeled with elements in Σ, an edge set $E(g)$ and a set $H(g)$ of variables. The set of all cograph patterns is denoted by $\mathcal{CGP}$. A cograph pattern having no variables is called a *cograph*. The set of all cographs is denoted by $\mathcal{CG}$. In this paper, we deal with cograph patterns satisfying that all variables in each cograph pattern have mutually distinct variable labels in $\mathcal{X}$. In [2], it is shown that G is a cograph if and only if there is no subset U of $V(G)$ such that the induced subgraph $G[U]$ is isomorphic to the chain consisting of 4 vertices. Moreover, in [2], Corneil showed that, for any subset $U \subseteq V(G)$, the induced subgraph $G[U]$ is a cograph.

For two cograph patterns g and f, we write $g \cong f$ if g is isomorphic to f. Let g be a cograph pattern. For a vertex or a variable u in $V(g) \cup H(g)$, $N_g(u)$ denotes the set of all neighborhoods of u, i.e., $N_g(u) = \{v \mid \{u, v\} \in E(g)\}$. Let x be a variable label in $\mathcal{X}$ and f a cograph. The form x/f is called a *variable replacement* of x by f. A new cograph pattern $g\{x/f\}$ is obtained by replacing the variable h having the variable label x with f. In detail, by applying the variable replacement x/f to g, we can construct a new cograph pattern $g\{x/f\} = (V(g) \cup V(f), E', H(G) \cup H(f) - \{h\})$, where $E' = (E(g) \cup E(f) \cup \{\{u, v\} \mid u \in N_g(h), v \in V(f)\}) - \{\{u, h\} \mid u \in N_g(h)\}$. A *substitution* is a finite collection of variable replacements $[x_1/f_1, x_2/f_2, \ldots, x_n/f_n]$, where x_i's are mutually distinct variable labels in $\mathcal{X}$, f_i's are cographs. For a substitution $\theta = [x_1/f_1, x_2/f_2, \ldots, x_n/f_n]$, a new cograph pattern $g\theta$ is obtained by applying all variable replacements x_i/f_i in θ to g sequentially, i.e., $g\theta \cong (\cdots((g\{x_1/f_1\})\{x_2/f_2\})\cdots)\{x_n/f_n\}$. For example, in Fig. 1, cograph pattern G is obtained from g by substituting f_1 and f_2 for x and y, respectively.

Definition 2. (**Cograph pattern language**) For a cograph pattern $g \in \mathcal{CGP}$, the *cograph pattern language* of g, denoted by $L(g)$, is defined as the set $\{G \in \mathcal{CG} \mid G \cong g\theta$ for some substitution $\theta\}$.

3 Inductive Inference of Cograph Pattern Languages

For a class $\mathcal{C}$, Angluin [1] and Shinohara [5] showed that if $\mathcal{C}$ has finite thickness, and the membership problem and the minimal language problem for $\mathcal{C}$ are solvable in polynomial time then $\mathcal{C}$ is polynomial time inductively inferable from positive data. We consider the class $\mathcal{L}_{\mathcal{CGP}} = \{L(g) \mid g \in \mathcal{CGP}\}$ as a target of inductive inference. For a set S, $|S|$ denotes the number of elements in S.

It is easy to see that the following lemma holds, that is, for any nonempty finite set $S \subseteq \mathcal{CG}$, the cardinality of the set $\{L \in \mathcal{L}_{\mathcal{CGP}} \mid S \subseteq L\}$ is finite.

Lemma 1. *The class $\mathcal{L}_{\mathcal{CGP}}$ has finite thickness.*

By presenting a polynomial time matching algorithm for solving the following Membership Problem for $\mathcal{L}_{\mathcal{CGP}}$ in Sec. 3.1, we show Theorem 1.

Membership Problem for $\mathcal{L}_{\mathcal{CGP}}$
Instance: A cograph pattern $g \in \mathcal{CGP}$ and a cograph $G \in \mathcal{CG}$.
Question: Does $L(g)$ contain G?

```
Algorithm MATCHING-$\mathcal{CGP}(g, G)$; // $g$ : a cograph pattern, $G$ : a cograph;
output: "yes" or "no";
begin
  Construct a cotree pattern $T[g]$ of $g$;
  Construct a cotree $T[G]$ of $G$;
  output MATCHING-$\mathcal{CTP}(T[g], T[G])$
end.
```

Fig. 2. Algorithm Matching-$\mathcal{CGP}$

Theorem 1. *Given a cograph pattern $g \in \mathcal{CGP}$ and a cograph $G \in \mathcal{CG}$, Membership Problem for $\mathcal{L}_{\mathcal{CGP}}$ is solvable in $O(nN^{1.5})$ time, where $n = |V(g)| + |H(g)|$ and $N = |V(G)|$.*

A *minimally generalized cograph pattern* explaining a given set of cographs $S \subseteq \mathcal{CG}$ is a cograph g such that $S \subseteq L(g)$ and there is no cograph pattern g' satisfying that $S \subseteq L(g') \subsetneq L(g)$. By giving a polynomial time algorithm for solving the following MINL Problem for $\mathcal{L}_{\mathcal{CGP}}$ in Sec. 3.2, we show Theorem 2.

MINL Problem for $\mathcal{L}_{\mathcal{CGP}}$
Instance: A nonempty set of cographs $S \subseteq \mathcal{CG}$.
Question: Find a minimally generalized cograph pattern $g \in \mathcal{CGP}$ explaining S.

Theorem 2. *Given a nonempty set of cographs $S \subseteq \mathcal{CG}$, MINL Problem for $\mathcal{L}_{\mathcal{CGP}}$ is solvable in $O(|S|N_{min}^3 N_{max}^{1.5})$ time, where $N_{min} = \min_{G \in S} |V(G)|$ and $N_{max} = \max_{G \in S} |V(G)|$.*

Therefore, we have the following main result.

Theorem 3. *The class $\mathcal{L}_{\mathcal{CGP}}$ is polynomial time inductively inferable from positive data.*

3.1 A matching algorithm for cograph patterns

In Fig. 2, we give a polynomial time algorithm Matching-$\mathcal{CGP}$ solving Membership Problem for $\mathcal{L}_{\mathcal{CGP}}$. This algorithm reduces the membership problem for $\mathcal{CGP}$ to the membership problem for tree representations of cographs. A tree representation of a cograph pattern, called a cotree pattern, is defined as follows.

Definition 3. (**Cotree pattern**) A *cotree pattern* is a node-labeled unordered tree satisfying the following conditions. (1) A label of an internal node whose depth is an odd number (resp., even number) is "0" (resp., "1"). (2) A label of a leaf is in $\Sigma \cup \mathcal{X}$.

A cograph pattern G can be represented by a cotree pattern T as follows. The internal node of T labeled with "0" (resp., "1") represents a disjoint union operation (resp., disjoint union and complement operations for complementing each graph) and called a "0-node" (resp., "1-node"). A leaf labeled with an element in Σ represents a vertex of G. A leaf labeled with an element in $\mathcal{X}$, called a variable, represents a variable of G. A cotree pattern is a unique representation of a cograph pattern [2]. For a cograph pattern G, $T[G]$ denotes the cotree representation of G. For a variable node h, $parent(h)$ denotes the parent of h. A cotree pattern having no variable node is called a *cotree* simply.

Let f be a cotree pattern and h a variable of f with variable label $x \in \mathcal{X}$. Let g be a cotree pattern having r as its root. Then the form x/g is called a *binding* for x. A new cotree pattern $f\{x/g\}$ can be obtained by applying the binding x/g in the following way.

1. If g has only one leaf or $parent(h)$ is a "1-node", then remove h and identify r with $parent(h)$.
2. If $parent(h)$ is a "0-node" and r has exactly one child, then remove both h and r and identify the child of r with $parent(h)$.
3. Otherwise, remove h and attach r to $parent(h)$ as a child.

A substitution $\theta = [x_1/g_1, \ldots, x_n/g_n]$ is a finite collection of bindings such that for any i, j $(1 \leq i < j \leq n)$, the variable label x_i and x_j are distinct. The cotree pattern $f\theta$ is obtained by applying all bindings in θ to f sequentially.

We can show the following lemma for cotree patterns.

Lemma 2. *For a cograph pattern g and a substitution $\theta = [x_1/g_1, \ldots .x_n/g_n]$, there exists a substitution $\theta_T = [x_1/T[g_1], \ldots, x_n/T[g_n]]$ such that $T[g\theta] \cong T[g]\theta_T$ holds.*

From lemma 2, Membership Problem for $\mathcal{L}_{\mathcal{CGP}}$ is reduced to the following membership problem for cotree patterns. We show Lemma 3.

Instance: The cotree pattern $T[g]$ of a cograph pattern g and the cotree $T[G]$ of a cograph G,

Question: Is there a substitution θ such that $T[G] \cong T[g]\theta$ holds?

Lemma 3. *The membership problem for cotree patterns is solvable in $O(nN^{1.5})$ time, where $n = |V(g)| + |H(g)|$ and $N = |V(G)|$.*

We propose an algorithm Matching-$\mathcal{CTP}$ to solve the membership problem for cotree patterns in polynomial time, based on a polynomial time pattern matching algorithm for linear interval graph patterns [7]. Corneil et al. [3] showed that for a cograph pattern g, $T[g]$ can be constructed from g in linear time w.r.t. $|V(g)| + |H(g)|$ and $|E(g)|$. From Lemmas 2 and 3, we show Theorem 1.

3.2 An MINL algorithm for cograph patterns

In this section, we assume that $|\Sigma| = \infty$. Let g and f be cograph patterns. The algorithm MINL-$\mathcal{CGP}$ (Fig. 3) solves the MINL problem for $\mathcal{L}_{\mathcal{CGP}}$. The lines 4–7 extend a cograph patterns g by adding variables as much as possible while $S \subseteq L(g)$ holds (Fig. 4). The lines 11–12 tries to replace each variable in g with a labeled vertex if it is possible. We have the following lemma. We omit the proof. From the following lemma, we show Theorem 2.

Lemma 4. *Let $g \in \mathcal{CGP}$ be the output of the algorithm* MINL-$\mathcal{CGP}$ *for an input S. Let g' be a cograph pattern satisfying that $S \subseteq L(g') \subseteq L(g)$. Then $g' \cong g$.*

```
Algorithm MINL-$\mathcal{CGP}(S)$;  // $S$ : a set of cographs;
output:   $g$ : a minimally generalized cograph pattern;
begin
  Let $g$ be a cograph pattern with one variable;
  Let $g_1$ (resp., $g_2$) be a connected (resp. unconnected) cograph
  with two unmarked variables (Fig. 4);
  if $S$ contains both connected and unconnected cographs then output $g$;
  foreach unmarked variable $h$ having variable label $x$ in $g$ do
      if MATCHING-$\mathcal{CGP}(g\{x/g_1\}, G)$="yes" for $\forall G \in S$ then $g := g\{x/g_1\}$
      else if MATCHING-$\mathcal{CGP}(g\{x/g_2\}, G)$="yes" for $\forall G \in S$ then $g := g\{x/g_2\}$
      else mark $h$;
  Unmark all variables of $g$;
  Let $\Sigma(S)$ be the set of all single vertices whose labels appear in $S$;
  foreach unmarked variable $h$ having variable label $x$ in $g$ do
      if $\exists a \in \Sigma(S)$ s.t. MATCHING-$\mathcal{CGP}(g\{x/a\}, G)$="yes" for $\forall G \in S$
          then $g := g\{x/a\}$ else mark $h$;
  output $g$
end.
```

Fig. 3. Algorithm MINL-$\mathcal{CGP}$

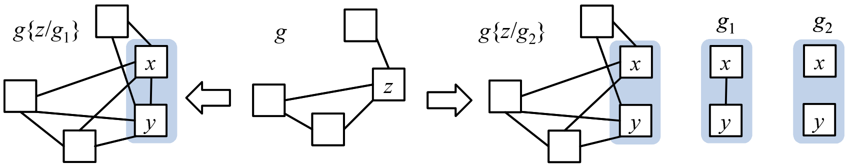

Fig. 4. Two refinement operators on Algorithm MINL-$\mathcal{CGP}$ (Fig. 3).

References

1. D. Angluin. Inductive Inference of Formal Languages from Positive Data. *Information and Control*, **45**, pp.117–135, 1980.
2. D.G. Corneil, H. Lerchs, and L. Stewart Burlingham. Complement Reducible Graph. *Discrete Applied Mathematics*, **3**, pp.163–174, 1981.
3. D.G. Corneil, Y. Perl, and L.K. Stewart. A Linear Recognition Algorithm for Cographs. *SIAM Journal on Computing*, **14(4)**, pp.926–934, 1985.
4. D. Lokshtanov, F. Mancini and C. Papadopoulos. Characterizing and Computing Minimal Cograph Completions. *Frontiers in Algorithmics, Second Annual International Workshop, FAW 2008, LNCS* **5059**, pages 147–158, 2008.
5. T. Shinohara. Polynomial Time Inductive Inference of Extended Regular Pattern Languages. *RIMS Symposia on Software Science and Engineering, LNCS*, **147**, pp.115–127, 1982.
6. R. Takami, Y. Suzuki, T. Uchida and T. Shoudai. Polynomial Time Inductive Inference of TTSP Graph Languages from Positive Data. *IEICE TRANSACTIONS on Information and Systems* , **E92-D(2)**, pp.181–190, 2009.
7. H. Yamasaki and T. Shoudai. A Polynomial Time Algorithm for Finding a Minimally Generalized Linear Interval Graph Pattern. *IEICE TRANSACTIONS on Information and Systems* **E92-D(2)**, pp.120–129, 2009.