Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol I,
IMECS 2010, March 17 - 19, 2010, Hong Kong

Learning Characteristic Structured Patterns in Rooted Planar Maps

Satoshi Kawamoto* Yusuke Suzuki† Takayoshi Shoudai*

Abstract—Extending the concept of ordered graphs, we propose a new data structure to express rooted planar maps, which is called a planar map pattern. In order to develop an efficient data mining method from a dataset of rooted planar maps, we propose a polynomial time algorithm for finding a minimally generalized planar map pattern, which represents maximal structural features common to rooted planar maps. Moreover, we show that the class of planar map patterns is polynomial time learnable from positive data.

Keywords: Planar Map, Graph Structured Pattern, Graph Mining, Machine Learning, Inductive Inference

1 Introduction

A *planar map* is a planar graph together with its embedding in the plane [7]. There are many applications of a geometric nature in which planar maps consisting of line and curve segments are occurred. In particular, measuring the similarity of patterns of planar maps is a standard problem in geographic information systems, CAGD (Computer Aided Geometric Design), computer vision, etc. For example, Alt et al. [1] gave an algorithm for geometrically measuring the similarity of two planar maps. Knowledge representations for planar map data structures are quite important to discover interesting features which such structured data have.

A *rooted planar map* is a planar map in which a single edge in the outer face is directed in a clockwise direction. The directed edge is called the root edge, and the vertex which is incident with the tail of the root edge is called the root vertex (or simply the root). Enumeration techniques of rooted or unrooted planar maps have been extensively studied [8]. There are several approaches to generate graphs for graph mining tasks [4, 6] and we have developed an efficient algorithm for generating all frequent outerplanar graph structured patterns by a refinement-based technique [9]. In this paper, we introduce a new topological data structure, called a *planar map pattern*, to represent common structural features of rooted planar maps. A planar map pattern is a graph structured pattern having a rooted planar map structure and structured variables. A variable can be replaced with arbitrary rooted planar maps. We say that a planar map pattern P matches a rooted planar map M if M is obtained from P by substituting rooted planar maps for all variables in P. The object of this paper is to propose an efficient algorithm for finding, given a set of rooted planar maps S, a planar map pattern which matches all elements in S and cannot be further specialized to S.

For a planar map pattern P, *the planar map pattern language of* P, denoted by $L(P)$, is the set of all rooted planar maps which are matched by P. For a set of rooted planar maps S, a planar map pattern P is said to be a *minimally generalized planar map pattern explaining* S if $L(P)$ contains all planar maps in S and is minimal among all planar map pattern languages which contain all planar maps in S. For considering learnabilities of planar map pattern languages, we use the framework of inductive inference. Angluin [2] and Shinohara [5] gave the framework of inductive inference from positive data and showed that if a concept class $\mathcal{C}$ has finite thickness, and the membership problem and the minimal language problem for $\mathcal{C}$ are computable in polynomial time then $\mathcal{C}$ is polynomial time inductively inferable from positive data. Based on this framework, in this paper, we consider the polynomial time learnabilities of *the class of planar map pattern languages* ($\mathcal{CPMPL}$).

Jiang and Bunke introduced the concept of ordered graphs and ordered graph isomorphism [3]. These graphs have the particularity of having all edges incident to a vertex uniquely ordered. For a rooted planar map or planar map pattern P, we assume that for every vertex v of P, the ordering of the edges in the adjacency list of v corresponds to the clockwise ordering of these edges around v in the embedding. By using this idea, we give two polynomial time algorithms for solving the following two problems: (1) the membership problem which is, given planar map M and planar map pattern P, to decide whether or not $L(P)$ contains M, and (2) the minimal language problem, MINL problem for short, which is, given a set of rooted planar maps S, to find a minimally generalized planar map pattern explaining S. Finally, we show that the class of planar map pattern languages is polynomial time inductively inferable from positive data.

*Department of Informatics, Kyushu University, 744 Motooka, Nishi-Ku, Fukuoka 819-0395, Japan. Email: {satoshi.kawamoto@i, shoudai@inf}.kyushu-u.ac.jp

†Graduate School of Information Science, Hiroshima City University, 3-4-1 Otsuka-Higashi, Asa-Minami-Ku, Hiroshima 731-3194, Japan. Email: y-suzuki@hiroshima-cu.ac.jp

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

2 Preliminaries

For a set or sequences r, the number of elements in r, called the size of r, is denoted by $|r|$. For a sequence $r = [r_1, r_2, \ldots, r_{|r|}]$, we denote $r[k] := r_k$ $(1 \le k \le |r|)$. Especially $r[0] := r[|r|]$. For two sequences $r = [r_1, r_2, \ldots, r_{|r|}]$ and $s = [s_1, s_2, \ldots, s_{|s|}]$, $r||s$ denotes the concatenation of r and s, i.e., $r||s := [r_1, r_2, \ldots, r_{|r|}, s_1, s_2, \ldots, s_{|s|}]$. For a sequence $r = [r_1, r_2, \ldots, r_{|r|}]$ and two integers i, j $(1 \le i < j \le |r|)$, we denote $r(i, j) := [r_i, r_{i+1}, \ldots, r_j]$.

In this paper, we treat a circularly-linked list as a sequence. Let r, s be circularly-linked lists at size k. We say that r and s are isomorphic, denoted by $r \cong s$, if there exists an integer t $(1 \le t \le k)$ such that $r(t, k)||r(1, t-1) = s$.

Definition 1 *Let $G = (V_G, E_G)$ be a plane graph with vertex set $V_G \subseteq R^2$ and edge set $E_G \subseteq V_G \times V_G$. We assume that G is biconnected. A rooted planar map M is defined as a triplet $M = (V_M, E_M, r_M)$, where $V_M = V_G$, $E_M = E_G$. r_M is one of the vertices which lie on the outer boundary, called the root.*

Let $M = (V_M, E_M, r_M)$ be a rooted planar map. The *outer frame* of M, denoted by R_M, is a list of vertices visited in clockwise order around the outer boundary starting from the root. A vertex in R_M is called an outer frame vertex. The rank of the outer frame vertex is defined as the order of R_M, where the rank of the root is 1. For a vertex v, if v is an outer frame vertex then $rank(v)$ is the rank of v, otherwise $rank(v) = 0$. We say that an edge $(u, v) \in E_M$ is an outer frame edge if $u, v \in R_M$. For a vertex $v \in V_M$, the *neighbors* of v, denoted by $L_M(v)$, is the circularly-linked list of adjacent vertices of v sorted in clockwise order. L_M is the set of the neighbors of all vertices in M i.e., $L_M = \{L_M(v) \mid v \in V_M\}$. The faces of M are the regions bounded by edges. In this paper, we treat a face as a circularly-linked list of vertices sorted in clockwise order. Let (u, v) be an edge of M, the face induced by (u, v) is the face such that its circularly-linked list includes vertices u, v in this order. In Fig. 1, we give an example of the rank of the outer frame vertices and the face induced by an edge.

Definition 2 *Let $M_1 = (V_1, E_1, r_1)$ and $M_2 = (V_2, E_2, r_2)$ be rooted planar maps. Let R_1, R_2 be outer frames of M_1, M_2 respectively and L_1, L_2 the sets of neighbors of M_1, M_2 respectively. We say that M_1 and M_2 are equivalent if there is a bijection ϕ from V_1 to V_2 satisfying the following conditions 1–4:*

1. *$(u, v) \in E_1$ if and only if $(\phi(u), \phi(v)) \in E_2$.*
2. *$L_1(v) = [u_1, u_2, \ldots, u_k]$ if and only if $L_2(\phi(v)) = [\phi(u_1), \phi(u_2), \ldots, \phi(u_k)]$.*

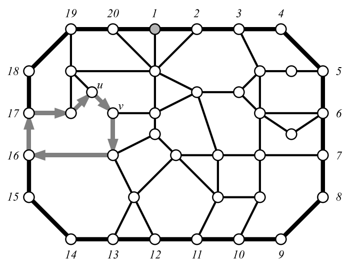

Figure 1: The rank of the outer frame vertex and the face induced by (u, v)

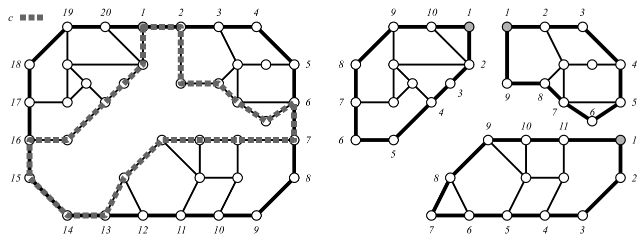

Figure 2: An example of a decomposition of a rooted planar map

3. *$\phi(r_1) = r_2$ i.e., the root of M_1 is mapped to the root of M_2.*
4. *$R_1 = [u_1, u_2, \ldots, u_k]$ if and only if $R_2 = [\phi(u_1), \phi(u_2), \ldots, \phi(u_k)]$*

Definition 3 *Let c be a face of $M = (V_M, E_M, r_M)$. The decomposition of M by c, denoted by $\overset{c}{M}$, is a sequence of rooted planar maps obtained from M by deleting outer frame edges in c. The root of each rooted planar map is defined as the outer frame vertex whose rank is least in these outer frame vertices. The ordering of these rooted planar maps in $\overset{c}{M}$ is defined in ascending order of the rank of the roots of these rooted planar maps.*

For any face in c in M, we compute the decomposition of M by c in $O(|V|)$ times. In Fig. 2, we give an example of a decomposition of a rooted planar map.

Definition 4 *Let $M = (V_M, E_M, r_M)$ be a rooted planar map. A planar map pattern P is defined as a 4-tuple (V_P, E_P, A_P, r_P) where*

- *$V_P = V_M$, $E_P = E_M$, and $r_M = r_P$.*
- *A_P is a subset of V_P. An element in A_P is called a variable.*
- *For $x \in A_P$, the vertices of $L_P(x)$ consists a cycle.*

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- *There does not exist a vertex which is adjacent to two or more variables.*

For a planar map pattern P and a variable x in P, the variable face of x is the neighbors of x. In this paper, a variable is drawn by a filled square.

A planar map pattern with no variable is called a *ground planar map pattern* and considered to be a rooted planar map. For a vertex v of a planar map pattern, the degree of v is the number of edges incident to v. The degree of v is denoted by $deg(v)$.

We use the following notation to describe our algorithm. Let $P = (V_P, E_P, A_P, r_P)$ be a planar map pattern and v a vertex in A_P. Let $L_P(v) = [u_1, u_2, \ldots, u_\ell]$ be the neighbors of v. For two integers i, k $(0 \leq i, k \leq \ell)$, we define

$$\mathrm{NGB}_P^v(u_i, k) = \begin{cases} u_{i+k} & \text{if } i + k \leq \ell, \\ u_{i+k-\ell} & \text{otherwise;} \end{cases}$$

$$\mathrm{NGB}_P^v(u_i, -k) = \begin{cases} u_{i-k} & \text{if } i - k \geq 1, \\ u_{i-k+\ell} & \text{otherwise;} \end{cases}$$

When the context is clear, the subscript P on the $\mathrm{NGB}_P^u(u_i, k)$ is dropped.

Let $P = (V_P, E_P, A_P, r_P)$ and $M = (V_M, E_M, A_M, r_M)$ be planar map patterns and $R_M = [u_0, u_1, \ldots, u_\ell]$ the outer frame of M. Let x be a variable in P with $L_P(x) = [v_0, v_1, \ldots, v_\ell]$. Then the form $x := M$ is called a binding for x. A new planar map pattern P' is obtained by applying the binding $x := M$ to P in the following way. We attach M to P by removing the variable x from A_P and by identifying the vertices $v_0, v_1, \ldots, v_\ell$ with the vertices $u_0, u_1, \ldots, u_\ell$ of M in this order.

A *substitution* θ is a finite collection of bindings $\{x_1 := M_1, x_2 := M_2, \ldots, x_n := M_n\}$. The planar map pattern $P\theta$, called the instance of P by θ, is obtained by applying all the bindings $\{x_i := M_i\}$ in θ to P.

In Fig. 3, we give an example of a substitution for a planar map pattern. Consider the example in Fig. 3. There is a substitution $\theta = \{x_1 := M_1, x_2 := M_2\, x_3 := M_3\}$. The planar map pattern $P\theta$ is equivalent to the rooted planar map in Fig. 1.

We say that a planar map pattern P matches a rooted planar map M if M is obtained from P by substituting rooted planar maps for all variables in P.

Definition 5 *For a planar map pattern P, the planar map pattern language of P, denoted by $L(P)$, is the set of all rooted planar maps which is matched by P.*

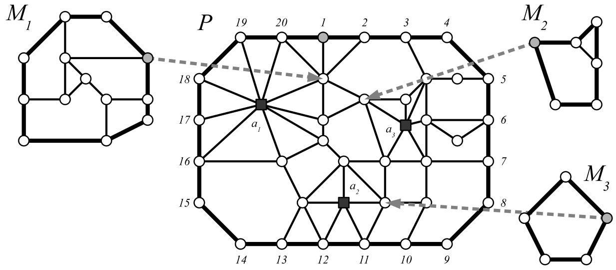

Figure 3: An example of substitutions for a planar map pattern

The class of planar map pattern languages ($\mathcal{CPMPL}$) is the set of all planar map pattern languages of all planar map patterns. We say that a planar map pattern P *explains* a given set of rooted planar maps S if $L(P)$ contains all rooted planar maps in S. A minimally generalized planar map pattern P explaining S is a planar map pattern P such that P explains S and $L(P)$ is minimal among all planar map pattern languages which contain all rooted planar maps in S. For a set of rooted planar maps S, there are possibly many minimally generalized planar map patterns explaining S in general.

3 A Polynomial Time Algorithm for Solving the Membership Problem

In this section, we give a polynomial time matching algorithm for solving the following membership problem.

Membership Problem for $\mathcal{CPMPL}$
Input: A rooted planar map M and a planar map pattern P.
Problem: Decide whether or not $M \in L(P)$.

To solve the membership problem, we give Algorithm Matching in Fig. 4. At first, this algorithm visits all vertices of the input planar map pattern except variables. Next, the algorithm visits the input rooted planar map in the same order as in the planar map pattern.

Lemma 1 *Algorithm* Matching *(Fig. 4) decides whether a planar map pattern P matches a rooted planar map M.*

Proof. From Def. 4, there does not exist a vertex which is adjacent to two or more variables, and the neighbors of a variable consists a cycle. Therefore the planar map pattern P' obtained from P by deleting all variables in P is biconnected. Hence Algorithm Matching is able to visit all vertices of P'. If the algorithm visits all vertices of M in the same order as in P', then P matches M.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Algorithm MATCHING(M, P);
input $M = (V_M, E_M, r_M)$: a rooted planar map, $P = (V_P, E_P, A_P, r_P)$: a planar map pattern;
output "*true*" or "*false*";
begin
  Let R_M be the outer frame of M and L_M the set of neighbors of M;
  Let R_P be the outer frame of P and L_P the set of neighbors of P;
  $P := (V_P \cup \{\varepsilon\}, E_P \cup \{(v_P, \varepsilon)\}, A_P, r_P)$ where ε lies on the outer boundary;
  $M := (V_M \cup \{\varepsilon'\}, E_M \cup \{(v_M, \varepsilon')\}, r_M)$ where ε' lies on the outer boundary;
  $c := 1$;
  Let $OPEN_P$ be a list of edges of P initialized to be $[(v_P, \varepsilon)]$;
  Let $OPEN_M$ be a list of edges of M initialized to be $[(v_M, \varepsilon')]$;
  while $OPEN_P$ is not empty **do begin**
    Let $OPEN_P[1] = (u, w)$; $OPEN_P := OPEN_P[2, \ldots]$;
    Let $OPEN_M[1] = (u', w')$; $OPEN_M := OPEN_M[2, \ldots]$;
    if u is adjacent to a variable x **then begin**
      Let $\mathrm{NGB}_P^w(u, p) = \mathrm{NGB}_P^w(u, -q) = x$;
      if $\deg(u') < p + q$ **then**
        return *false*;
      for $i := 1$ to $p - 1$ **do begin**
        if $NGB_P^w(u, i)$ is not attached any label **then begin**
          if the label of $NGB_P^w(u, i)$ is not equal to the label of $NGB_M^w(u', i)$ **then** return *false*
        end else if $OPEN_P$ does not contain $NGB_P^w(u, i)$ **then begin**
          $OPEN_P := OPEN_P \& [(NGB_P^w(u, i), u)]$;
          $OPEN_M := OPEN_M \& [(NGB_M^w(u', i), u)]$
        end
      end;
      for $i := 1$ to $q - 1$ **do begin**
        if $NGB_P^w(u, -i)$ is not attached any label **then begin**
          if the label of $NGB_P^w(u, -i)$ is not equal to the label of $NGB_M^w(u', -i)$ **then** return *false*
        end else if $OPEN_P$ does not contain $NGB_P^w(u, -i)$ **then begin**
          $OPEN_P := OPEN_P \& [(NGB_P^w(u, -i), u)]$;
          $OPEN_M := OPEN_M \& [(NGB_M^w(u', -i), u)]$
        end
      end
    end else begin
      if $\deg(u') \neq \deg(u)$ **then** return *false*;
      for $i := 1$ to $\deg(u) - 1$ **do begin**
        if $NGB_P^w(u, i)$ is not attached any label **then begin**
          if the label of $NGB_P^w(u, i)$ is not equal to the label of $NGB_M^w(u', i)$ **then** return *false*
        end else if $OPEN_P$ does not contain $NGB_P^w(u, i)$ **then begin**
          $OPEN_P := OPEN_P \& [(NGB_P^w(u, i), u)]$;
          $OPEN_M := OPEN_M \& [(NGB_M^w(u', i), u)]$
        end
      end
    end;
    Attach label c to u and u';
    $c := c + 1$
  end;
  return *true*
end.

Figure 4: Algorithm MATCHING.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Thus Algorithm MATCHING correctly decides whether or not P matches M. □

Lemma 2 *Membership problem for $\mathcal{CPMPL}$ is solvable in polynomial time.*

Proof. The correctness follows from Lemma 1. Let $|E|$ be the number of edges of an input rooted planar map. At each iteration of while-loop, the algorithm visits a new vertex of the input rooted planar map and adds an edge to OPEN list. This operation repeated at most $O(|E|)$ times. Therefore the total time of Algorithm MATCHING (Fig. 4) is $O(|E|)$ time. □

4 A Polynomial Time Algorithm for Finding a Minimally Generalized Planar Map Pattern

In this section, we give a polynomial time algorithm for solving the following minimal language problem.

Minimal Language (MINL) Problem for $\mathcal{CPMPL}$
Input: A nonempty finite set of rooted planar maps S.
Output: A minimally generalized planar map pattern explaining S.

Algorithm MINL (Fig. 5) finds a minimally generalized planar map pattern explaining given a set of rooted planar maps all of whose lengths of the outer frame are equal.

Lemma 3 *For any two planar map patterns P_1 and P_2, $L(P_1) \subseteq L(P_2)$ if and only if P_1 is obtained from P_2 by substituting rooted planar maps for variables in P_2.*

Proof. The "if" part is obvious. We prove the "only if" part. Let k be an integer. We assume that k is more than the number of vertices of P_2. Let P_w be a ground term tree obtained from P by replacing all the variables in P_1 with rooted planar maps consisting of 2 faces one of which is bounded by k vertices. Since $P_w \in L(P_1)$ and $L(P_1) \subseteq L(P_2)$, $P_w \in L(P_2)$. Since a face bounded by k vertices does not appear in P_2, we have that P_2 is obtained from P_1 by inverting the substitutions. □

Lemma 4 *For an input set of rooted planar maps S, Algorithm* MINL *(Fig. 5) outputs a minimally generalized planar map pattern explaining S.*

Proof. Let M_r be a planar map pattern computed by Algorithm MINL (Fig. 5) for an input S. We show that if there exists a planar map pattern M_b such that $S \subseteq L(M_b) \subseteq L(M_r)$, M_r and M_b are equivalent. By Lemma 3, there exists a substitution θ such that $M_r\theta$ and M_b are equivalent. Algorithm MINL finds all common faces among the input rooted planar maps in S. Thus, each planar map patterns in binding in θ consists of only one variable face. Then, M_r and M_b are equivalent. □

Lemma 5 *Minimal Language Problem for $\mathcal{CPMPL}$ is computable in polynomial time.*

Proof. The correctness follows from Lemma 4. Let $S = \{M_1, \ldots, M_m\}$ be an input set of rooted planar maps, where $M_i = (V_i, E_i, r_i)$ $(1 \leq i \leq m)$ and R_i is the outer frame of M_i. Let $V_{\max} = \max_{1 \leq i \leq m} |V_i|$ and $\ell = \max_{1 \leq i \leq m} |R_i|$. Procedure MINL-SUB is called recursively at most $O(V_{\max})$ times in all. The procedure needs totally $O(\ell \times m \times V_{\max})$ time. Hence the total time for all executions in Algorithm MINL is $O(\ell \times m \times V_{\max}^2)$, which is polynomial w.r.t. S. □

5 Polynomial Time Learnability of Planar Map Pattern Languages

In this section, we give a theoretical result for learning planar map pattern languages. Angluin [2] and Shinohara [5] showed the following theorem.

Theorem 1 (Angluin, Shinohara) *For a class $\mathcal{C}$, if $\mathcal{C}$ has finite thickness, and the membership problem and the MINL problem for $\mathcal{C}$ are computable in polynomial time then $\mathcal{C}$ is polynomial time inductive inferable from positive data.*

It is easy to see that the following lemma holds, that is, for any nonempty finite set of rooted planar maps S, the cardinality of the set $\{L \mid S \subseteq L\}$ is finite.

Lemma 6 *The class $\mathcal{CPMPL}$ has finite thickness.*

Proof. Let S be a nonempty finite set of rooted planar maps and $M = (V_M, E_M, r_M)$ a rooted planar map in S. If $P = (V_P, E_p, A_P, r_P)$ is a planar map pattern such that $L(P)$ includes M, then $|V_P| \leq |V_M|$, $|A_P| \leq 3|V_P|$ and $|E_P| \leq |E_M|$. Hence $|\{P \mid M \subseteq L(P)\}|$ is finite. Since $\{P \mid S \subseteq L(P)\} = \bigcup_{M \in S}\{P \mid M \subseteq L(P)\}$, $|\{L(P) \mid S \subseteq L(P)\}|$ is finite. Therefore the class $\mathcal{CPMPL}$ has finite thickness. □

From Lemmas 2, 5, 6, and Theorem 1, we have the following theorem.

Theorem 2 *The class $\mathcal{CPMPL}$ is polynomial time inductively inferable from positive data.*

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Algorithm MINL(M);
input M : array[1. . . ,m] of rooted planar maps;
output a minimally generalized planar map pattern explaining M;
begin
  Let $P := M[1]$ be a temporary planar map pattern;
  P:=MINL-SUB(M, P);
  return P
end.

Procedure MINL-SUB(M, P);
input M : array[1. . . ,m] of rooted planar maps, P : a temporary planar map pattern;
begin
  for $t := 1$ to m **do** $c[t]$:= the outer frame of $M[t]$;
  Let ℓ be the length of outer frame;
  for $i := 1$ to ℓ **do begin**
    for $j := 1$ to m **do begin**
      $d_j = [v_0, v_1, \ldots, v_n]$ is the face induced by $(c[j][i-1], c[j][i])$ of $M[j]$;
      Let $s_j = [rank(v_0), rank(v_1), \ldots, rank(v_n)]$.
    end;
    if $s_1 = s_2 = \ldots = s_m$ and $s_i \neq \emptyset$ $(1 \leq i \leq m)$ **then begin**
      for $k := 1$ to $\left|\overset{d_1}{M}[1]\right|$ **do** MINL-SUB$\left(\left[\left(\overset{d_1}{M}[1]\right)[k], \left(\overset{d_2}{M}[2]\right)[k], \ldots, \left(\overset{d_m}{M}[m]\right)[k]\right], P\right)$;
      return P
    end
  end;
  Add a new variable x in the face $c[1]$ of P, and add edges from x to the all vertices of $c[1]$.
end;

Figure 5: Algorithm MINL.

6 Conclusions

In this paper, we gave a polynomial time learning algorithm for finding a characteristic structured patterns in rooted planar maps. More precisely we showed that the class of planar map pattern languages is polynomial time inductively inferable from positive data. Even if variables with the same variable label occur more than once, we can give polynomial time algorithms for both membership and MINL problems. A planar map pattern represents an expressive graph structure embedded in the plane. We are now developing graph mining methods for generating all frequent planar map patterns by a refinement-based technique proposed in this paper.

References

[1] H. Alt, A. Efrat, G. Rote, and C. Wenk, Matching Planar Maps, J. Algorithms 49(2), 262–283, 2003.

[2] D. Angluin, Inductive Inference of Formal Languages from Positive Data, Information and Control 45, 117–135, 1980.

[3] X. Jiang and H. Bunke, Marked Subgraph Isomorphism of Ordered Graphs, Proc. Joint IAPR Int. Workshops SSPR '98 / SPR '98, LNCS 1451, Springer, 122–131, 1998.

[4] M. Kuramochi and G. Karypis, Finding frequent patterns in a large sparse graph, Data Min. Knowl. Discov. 11, 243–271, 2005.

[5] T. Shinohara, Polynomial Time Inference of Extended Regular Pattern Languages. Proc. RIMS Symp. on Soft. Sci. and Eng., LNCS 147, Springer, 115–127, 1982.

[6] J. Ramon and S. Nijssen, Polynomial-Delay Enumeration of Monotonic Graph Classes, J. Machine Learning Research 10, 907–929, 2009.

[7] W.T. Tutte, Graph Theory, Cambridge Mathematical Library, 2001.

[8] T.R. Walsh, A biased survey of map enumeration results, Proc. the 10th WSEAS Int. Conf. on Applied Mathematics, 325–331, 2006.

[9] H. Yamasaki, Y. Sasaki, T. Shoudai, T. Uchida, and Y. Suzuki, Learning block-preserving graph patterns and its application to data mining, Machine Learning 76(1), 137–173, 2009.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)