Polynomial Time Pattern Matching Algorithm for Ordered Graph Patterns

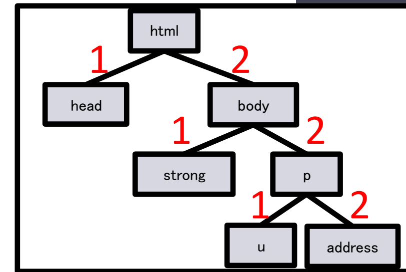
Takahiro Hino[1], Yusuke Suzuki[1],
Tomoyuki Uchida[1], Yuko Itokawa[2]

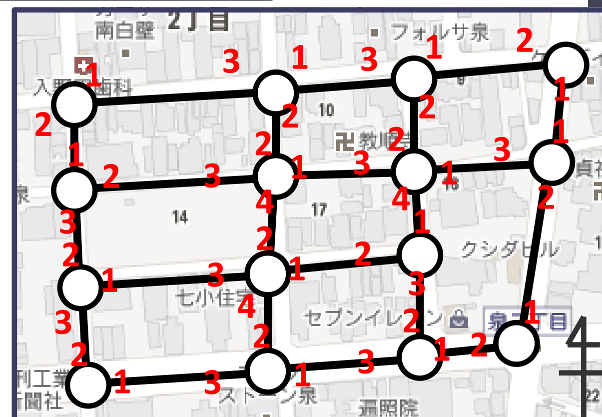
[1]Department of Intelligent Systems, Hiroshima City University, Japan
[2]Faculty of Psychological Science, Hiroshima International University, Japan

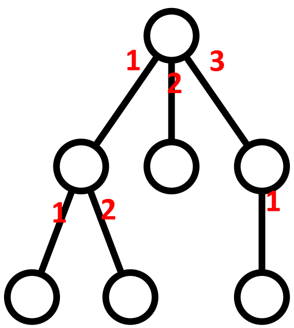
Background

Structured data such as Web pages, TEX sources, CAD and MAP are modeled by graphs each of whose vertices has ordered neighbors.

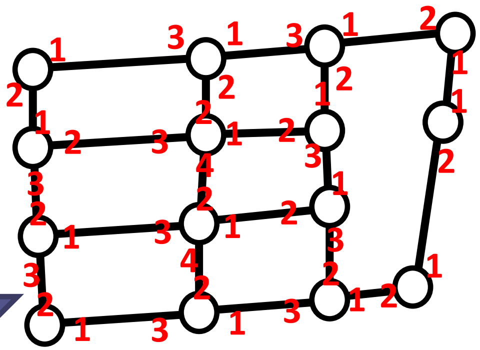
```
<html>
  <head>
    <title>Taro Hiroshima</title>
  </head>
  <body>
    <strong>Taro Hiroshima</strong>
    <p>
      <u>address</u>
      <address>taro@hiroshima-u.ac.jp</address>
    </p>
  </body>
</html>
```



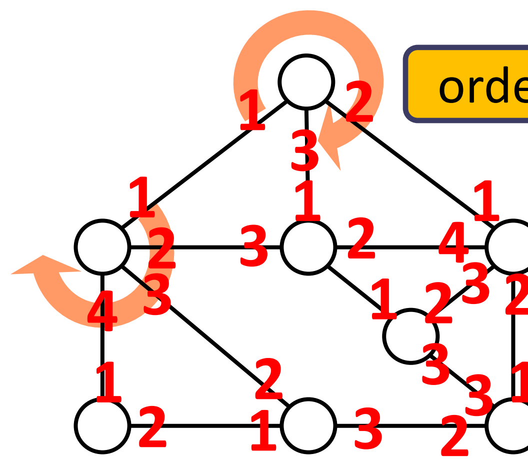




ordered tree



Planar map

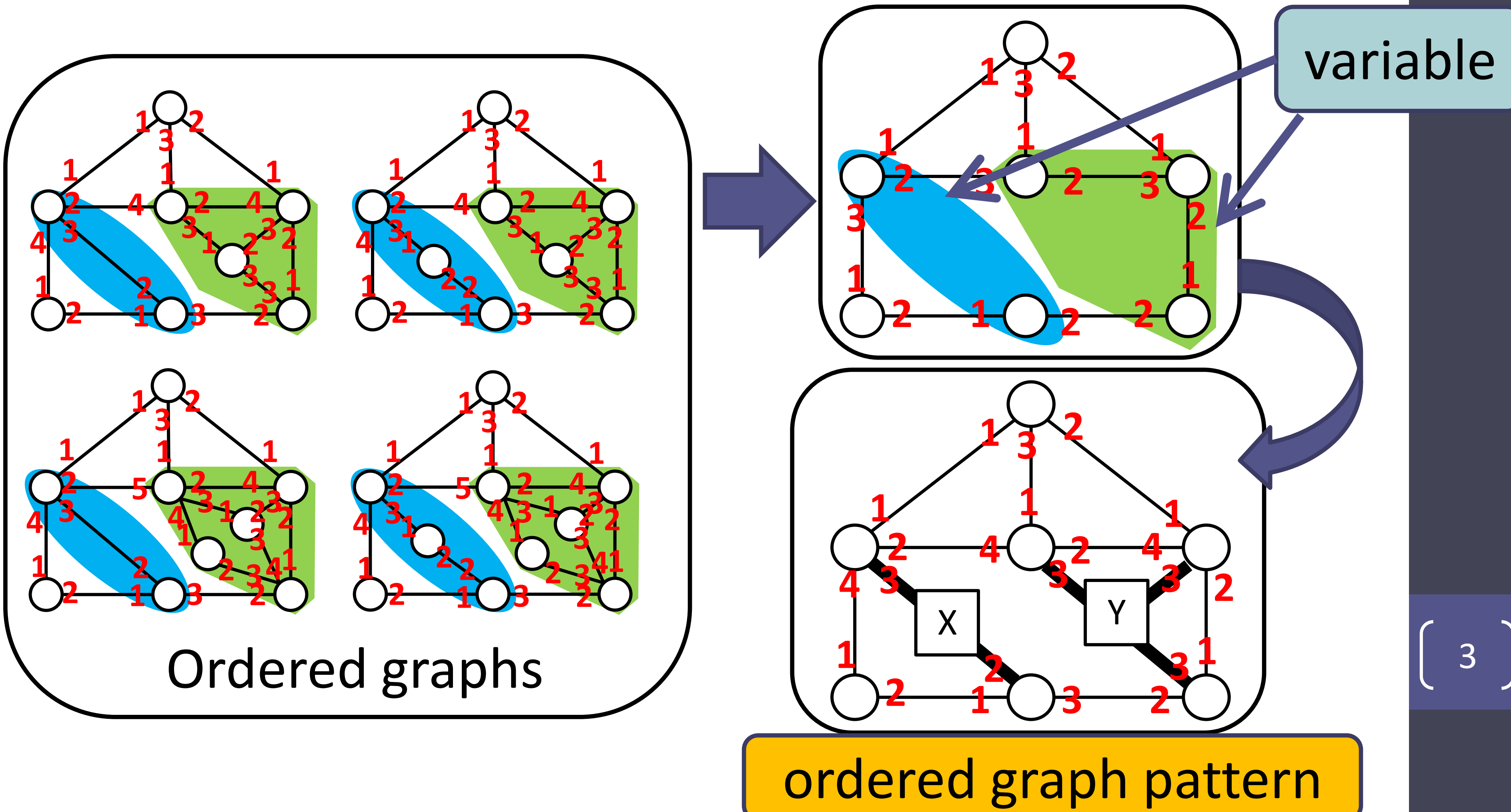


ordered graph

Ordered graph[Jiang and Bunke,1998]: Each vertex has a unique order based on neighboring vertices.

Motivation

Purpose of this research is to discover characteristic ordered graph patterns common to ordered graphs.



Our approach

1. **Ordered graph patterns** as Knowledge representation (This paper)

Ordered graph pattern has structural variables and each of whose vertices has an order on neighboring vertices and structural variables.

2. **Hypothesis checking process** for knowledge discovery (This paper)

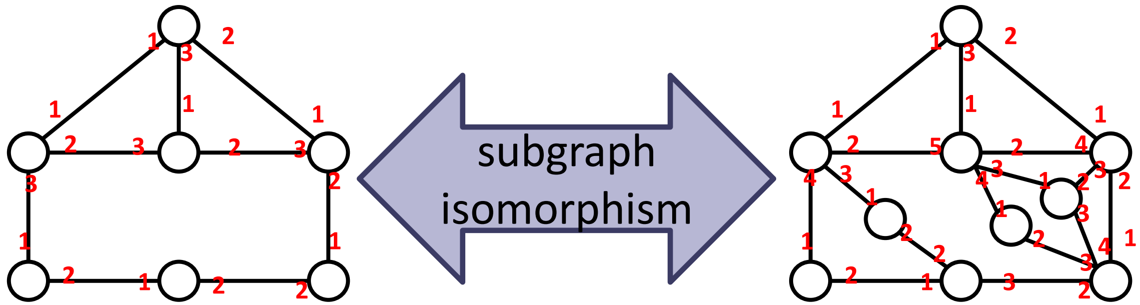
We present an efficient matching algorithm to solve the following problem.
Membership problem:
Does an ordered graph pattern g represent an ordered graph G?

3. **Learning method** from ordered graphs

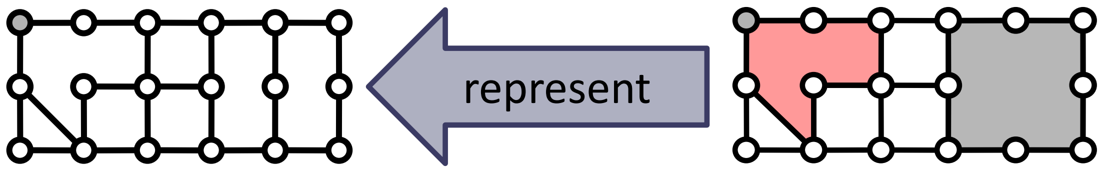
e.g. Inductive inference from positive data

Related works

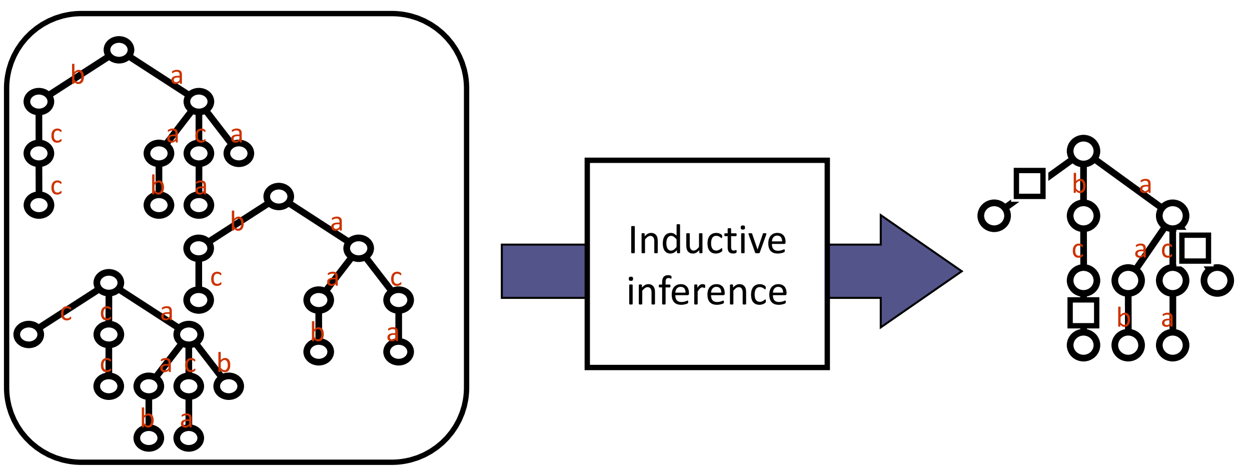
1. Ordered subgraph isomorphism[X. Jiang and H. Bunke,1998]



2. Matching algorithm for planar map[S. Kawamoto et al.,2010]



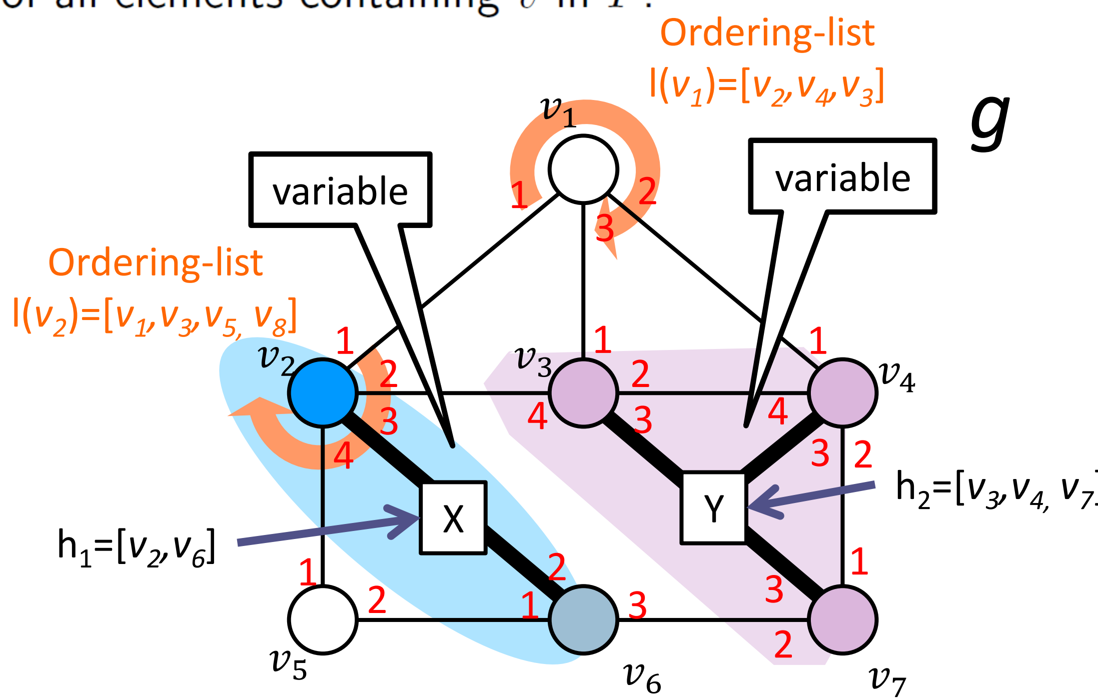
3. Inductive inference of ordered term tree[Y. Suzuki et al., 2006]



2.Preliminaries: Ordered graph pattern

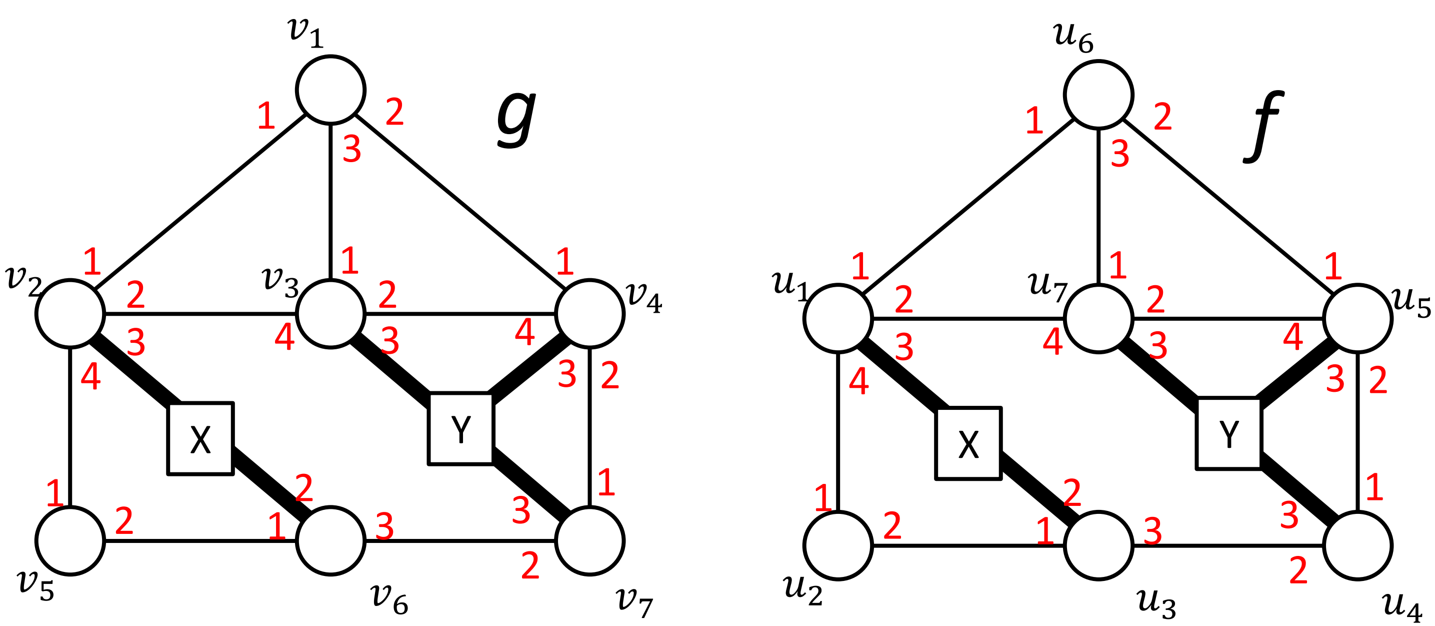
An ordered graph pattern is a five-tuple $(V, E, H, F, \mathcal{L})$ defined as follows.

(1) (V, E, H) is a graph pattern.
(2) F is a multiset of elements in the set $\{\{u, v\} \mid u \in V, v \in (V \cup H)\}$, such that $F \supseteq \{\{u, v\} \mid (u, a, v) \in E\}$ and the graph pattern $(V, \{\{u, v\} \mid u, v \in V, \{u, v\} \in F\}, \emptyset)$ is connected. An element in F is called an ordering-edge.
(3) $\mathcal{L} = \{\ell_F(v) \mid v \in V\}$, called an ordering-set, where $\ell_F(v)$, called an ordering-list of u on F, is a cyclic list of all elements containing v in F.



Ordering isomorphic

An ordered graph pattern g is said to be ordering isomorphic to an ordered graph pattern f if there exists a bijection $\varphi : V(g) \rightarrow V(f)$ satisfying the following three conditions. (1) $(u, a, v) \in E(g)$ if and only if $(\varphi(u), a, \varphi(v)) \in E(f)$. (2) $(u_0, x, u_1, \ldots, u_k) \in H(g)$ if and only if $(\varphi(u_0), x, \varphi(u_1), \ldots, \varphi(u_k)) \in H(f)$. (3) $\ell_{F(g)}(v) = (v_1, \ldots, v_k) \in \mathcal{L}(g)$ if and only if $\ell_{F(f)}(\varphi(v)) = (\varphi(v_1), \ldots, \varphi(v_k)) \in \mathcal{L}(f)$, where for a variable $(w_1, \ldots, w_t) \in H(g)$, $\varphi((w_1, \ldots, w_t))$ is defined as $(\varphi(w_1), \ldots, \varphi(w_t))$.



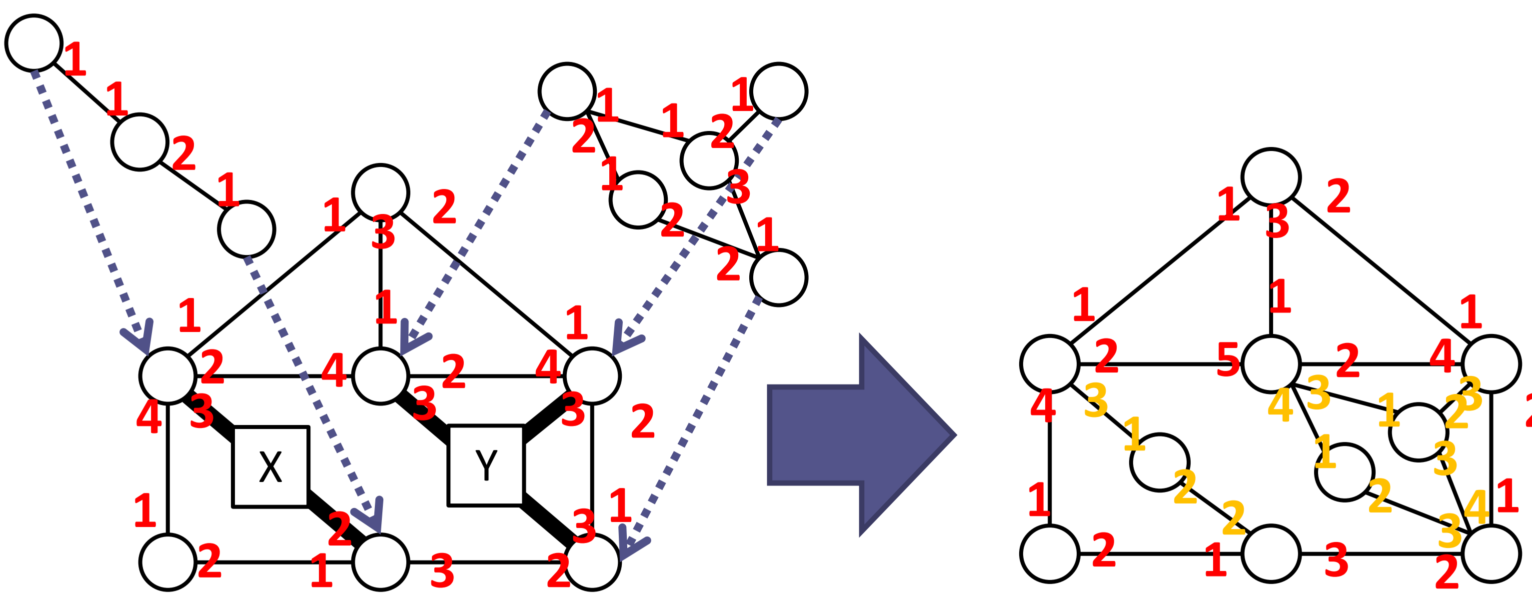
Lemma 1

For two ordered graph patterns g and f, a problem of determining whether or not g is ordering isomorphic to f is solvable in $O(\|\mathcal{L}(g)\|^2)$ time.

Substitution for an ordered graph pattern

Replace a variable with an ordered graph in the following way.

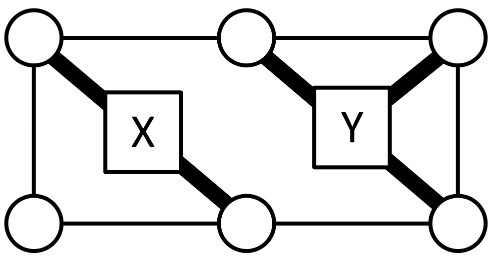
1. Remove the variable
2. Identify the port of variable with the vertices of ordered graph
3. Update ordering-list



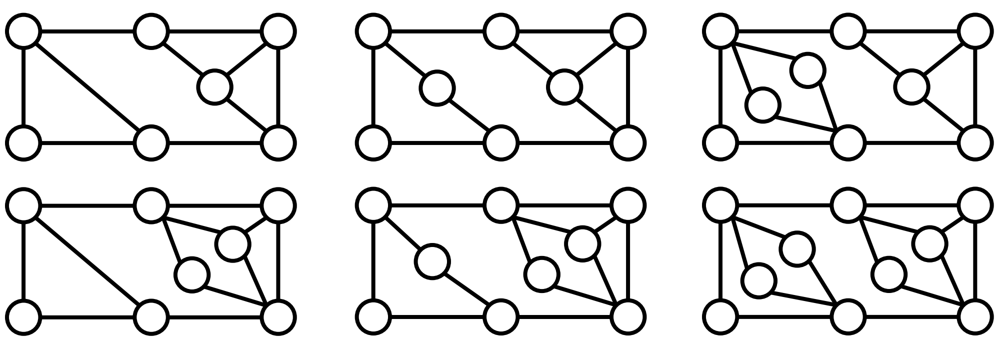
Ordered graph pattern language

For an ordered graph pattern *g*, the ordered graph pattern language of *g*, denoted by L(*g*), is defined as the set of all ordered graphs obtained from an ordered graph pattern by arbitrary substitutions.

g



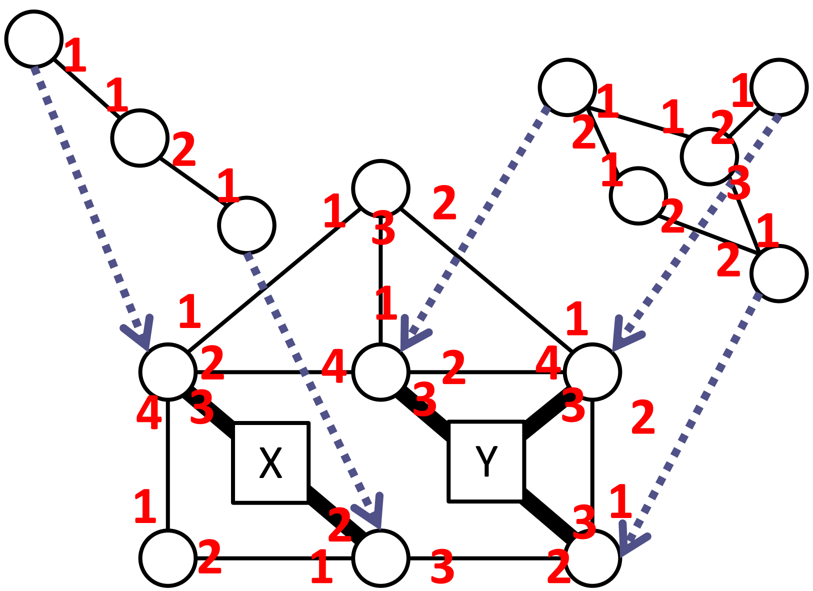
L(*g*)=



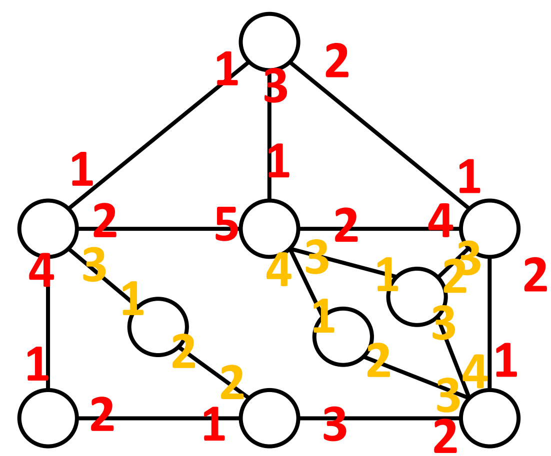
……

3. Pattern matching algorithm for Membership problem for ordered graph patterns

Instance： An ordered graph G and an ordered graph pattern *g*.
problem: Does L(*g*) contain G?



g



G

L(*g*) contains G

Matching algorithm

Input: ordered graph G, ordered graph pattern *g*.

begin

1. Let u be a vertex of *g* and {u, v} in cyclic list of u;
2. C :=Coding({u,v} *g*);
3. for each w in V (G) do
4. foreach e in cyclic list of w do
5. if CodeMatch(C,e,G) and StructureMatch (C,*g*,G)
6. then return true;
7. return false

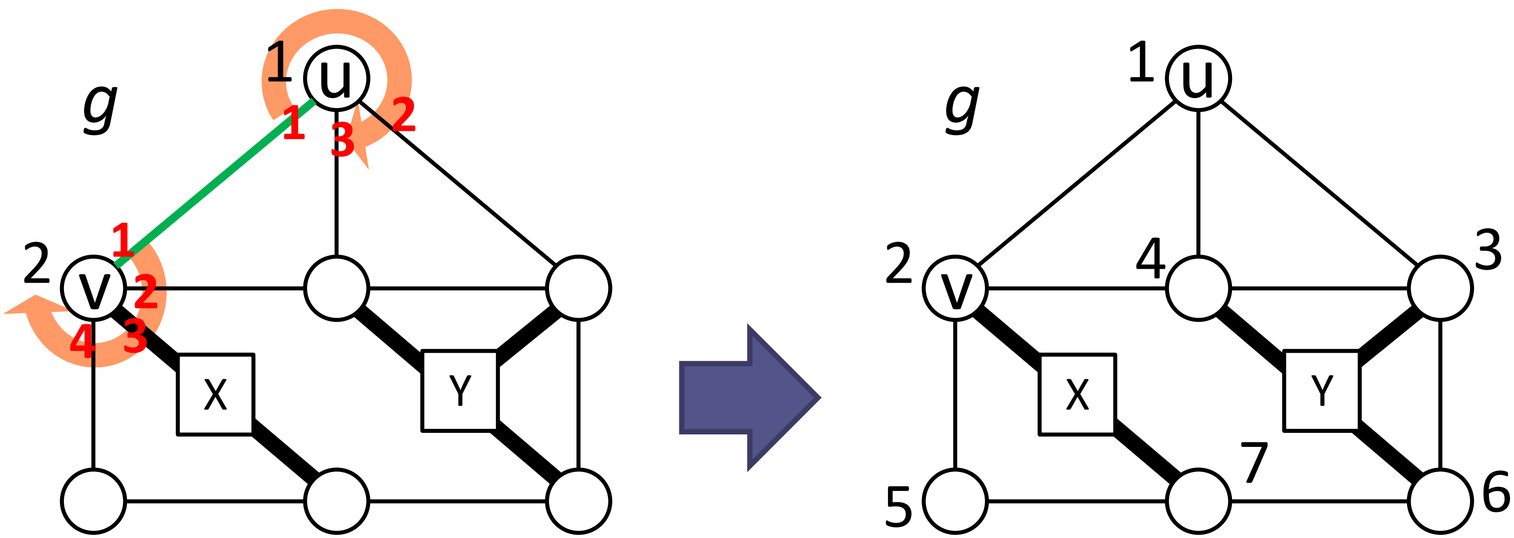
end.

Lemmas 2 and 3

For an ordered graph pattern g and an ordered graph G, g and G match on code and on graph structure if and only if g matches G.

Coding({u,v},*g*)

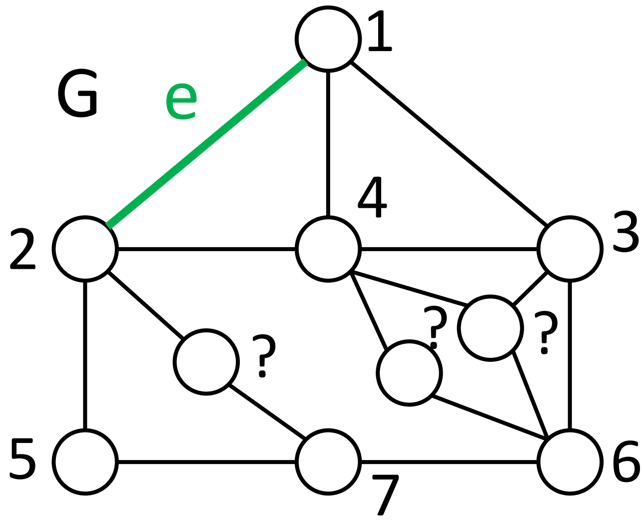
This procedure visits all vertices of the input ordered graph pattern g and attaches the ID of each vertex to the code of *g* by modifying polynomial time coding algorithm [X.Jiang and H.Bunke,1998] for ordered graphs.



C=#234#14x5#16y4#13y2#27#37y#5x6#

CodeMatch(C,e,G)

The procedure checks whether or not the code of *g* matches a code of G.



C=# 234#14x5#16y4#13y2#27#37y#5x6#

Code(G)=#234#14?5#16?4#13??2#27#37??#5?6#

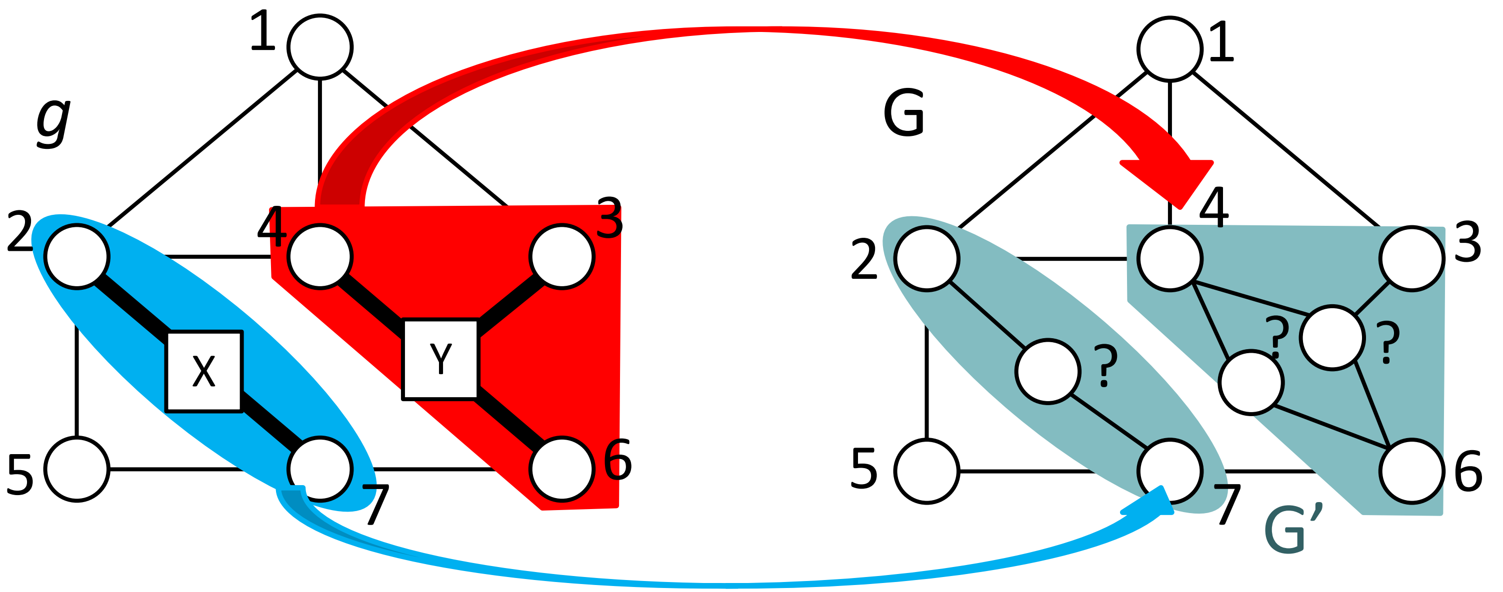
Replace $?^+$ with x or y

Code(G)=#234#14x5#16y4#13y2#27#37y#5x6#

C = Code(G) ⇒ CodeMatch(C,e,G) returns yes

StructureMatch(C,*g*,G)

The procedure StructureMatch determines whether or not *g* and G match on graph structure.



For an ordered graph pattern g and an ordered graph G such that g and G match on code, we say that g and G *match on graph structure* if the following conditions hold. Let $\varphi : V(g) \to V(G)$ be a one-to-one mapping obtained by $\textsc{Coding}(e, g, \emptyset)$.

(1) For each variable $(u_0, x, u_1, \ldots, u_k) \in H(g)$, all vertices $\varphi(u_1), \varphi(u_1), \ldots, \varphi(u_k)$ in $V(G)$ are in the same component of G'.

(2) For any two distinct variables $(u_0, x, u_1, \ldots, u_k), (v_0, x, v_1, \ldots, v_k) \in H(g)$ having the same variable label x, the component in G' having $\varphi(u_0), \varphi(u_1), \ldots, \varphi(u_k)$ is ordering isomorphic to the component in G' having $\varphi(v_0), \varphi(v_1), \ldots, \varphi(v_k)$.

Main Theorem

Theorem 1

For an ordered graph pattern *g* and on ordered graph G, the membership problem for *g* and G is solvable in polynomial time.

The total time of the algorithm is $O(|V(g)| \times \|\mathcal{L}(G)\|^2)$.

$|V(g)|$:The number of vertices of ordered graph pattern *g*.
$\|\mathcal{L}(G)\|$:The number of ordering-edges of ordered graph G.

Conclusion

1. We have proposed an ordered graph pattern as a new graph pattern.
2. We have proposed a polynomial time matching algorithm to solve the membership problem for ordered graph patterns.

Future Work (1)

1. It can be shown that the class of ordered graph pattern languages is polynomial time inductively inferable from positive data by showing the following three propositions.
   1. The class of ordered graph pattern languages has finite thickness. --- obviously hold
   2. The membership problem of ordered graph pattern languages is solvable in polynomial time. --- showed in this paper.
   3. The minimal language problem of ordered graph pattern languages is solvable in polynomial time. --- future work

Future Work (2)

2. For graph structured data such as Web pages, TEX files, CAD, MAP, we are designing a graph mining tools having the following knowledge representations.
   1. Ordered graph pattern representing graph structures common to graph structured data. --- this work

      We are implementing a graph mining tools having an ordered graph pattern as a knowledge representation.
   2. Layout Formal Graph System [Uchida et al., 2000] which is a first-order logic programming system dealing with ordered graph pattern as a term.

      ----- Extension of this presentation