+

Learning Unordered Tree Contraction Patterns in Polynomial Time

Yuta Yoshimura and Takayoshi Shoudai

Department of Informatics, Kyushu University, Japan

Outline

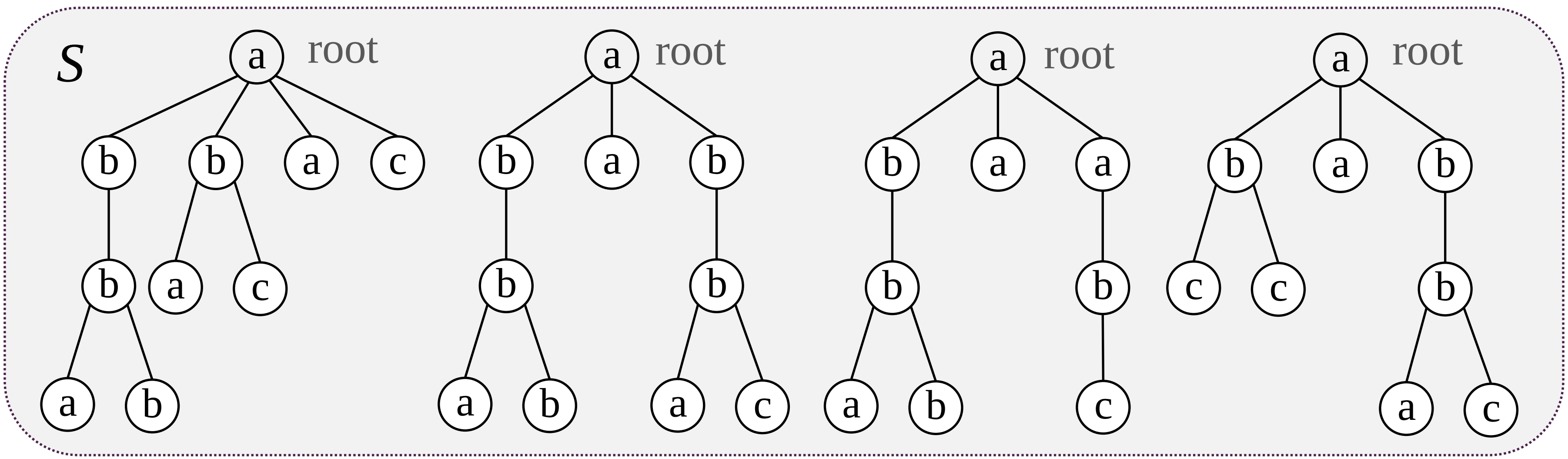
1. Backgrounds & motivations
2. Preliminaries
   - Tree contraction pattern (TC-pattern)
3. Time complexity of the TC-pattern matching problem
4. The minimal language problem for TC-patterns
5. Conclusions and future work

Backgrounds & motivations

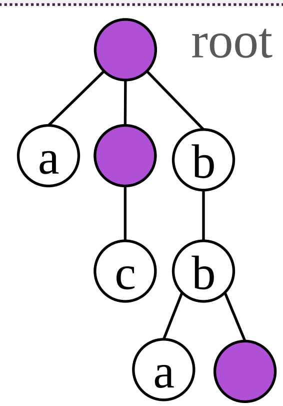
- Increase of tree-structured data
  - Web documents
  - XML files etc..
- Discovery common characteristic tree-structured patterns from tree-structured database
- Application
  - Classification of a tree-structured data set

Backgrounds & motivations

- A graph pattern expression common to given tree structured data



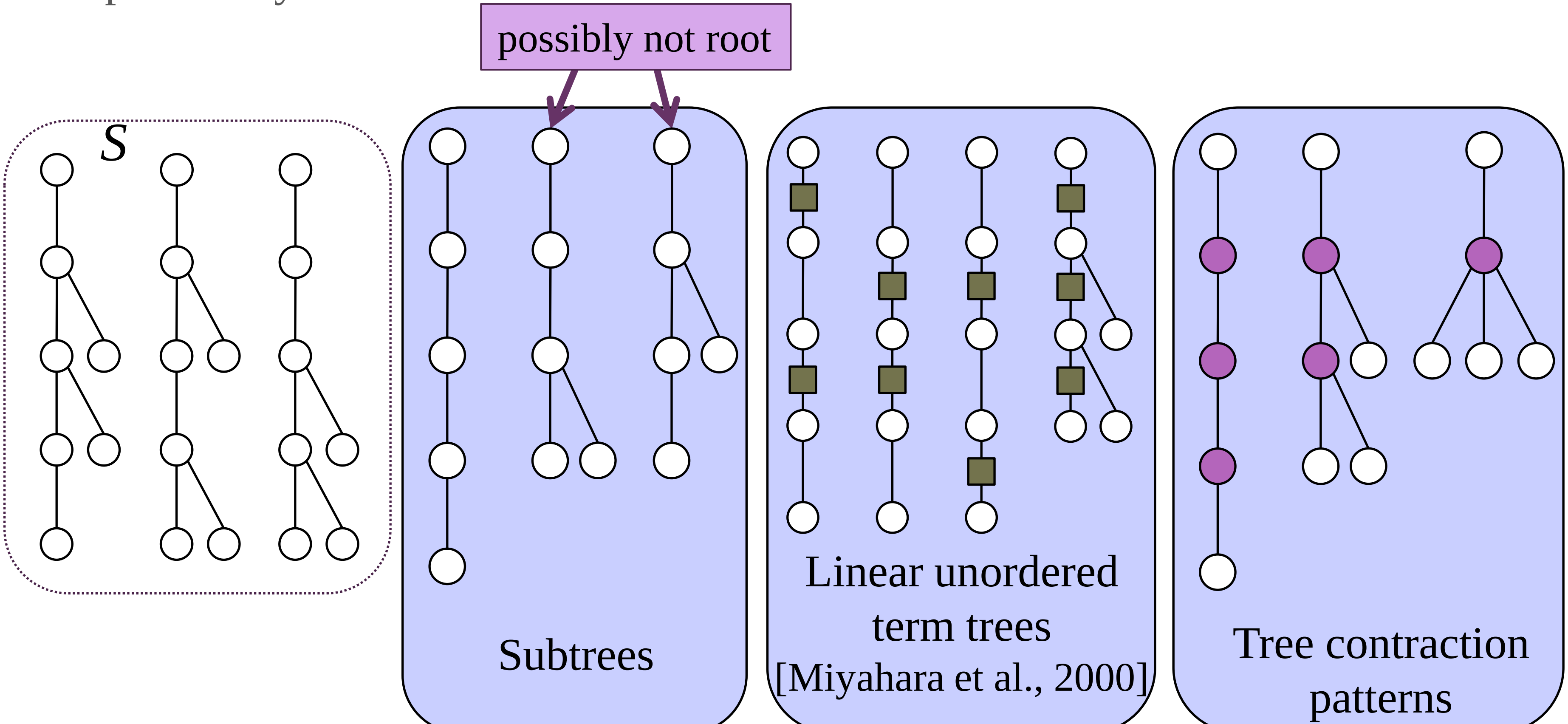
A common pattern



In this talk, the top vertex of each trees is always it's root.

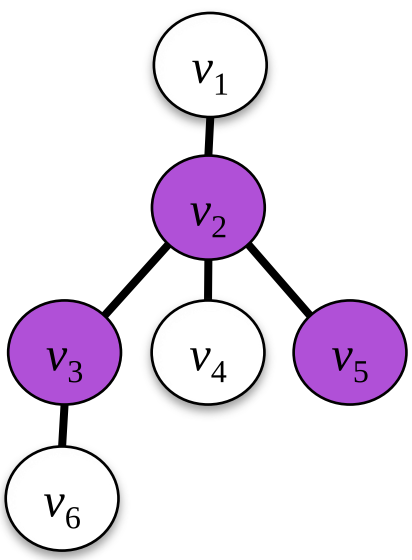
Backgrounds & motivations

- A difference between tree contraction patterns and term trees which previously studied.



Tree contraction pattern (TC-pattern)

- A *tree contraction pattern* (*TC-pattern*) is a triplet $t = (V_t, E_t, U_t)$ where
  - V_t is a vertex set,
  - E_t is an edge set, and
  - U_t is a subset of V_t, whose elements are called contractible vertices. Below, purple vertices indicate contractible vertices.
  - (V_t, E_t) is a tree with a specified root $r_t \in V_t$.



TC-pattern t

$V_t = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

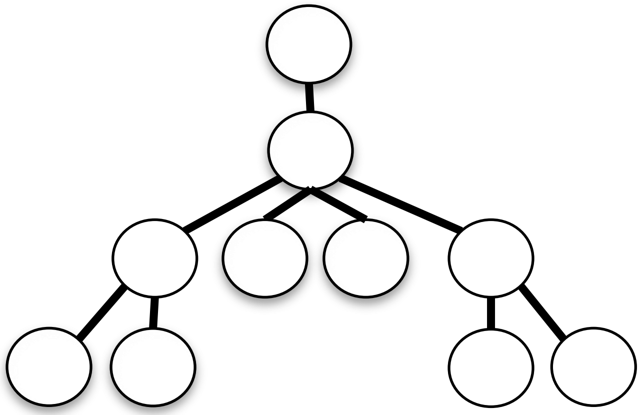
$E_t = \{\{v_1,v_2\},\{v_2,v_3\},\{v_2,v_4\},\{v_2,v_5\},\{v_3,v_6\}\}$

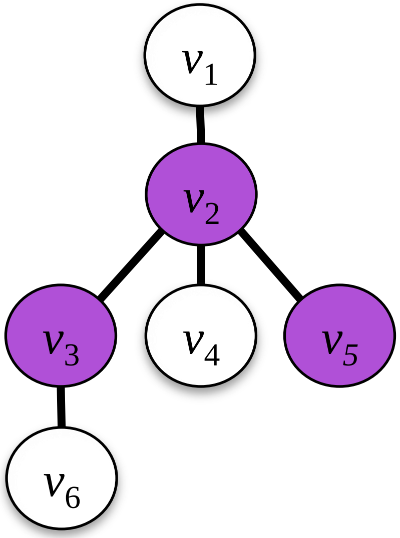
$U_t = \{v_2, v_3, v_5\}$

$r_t = v_1$

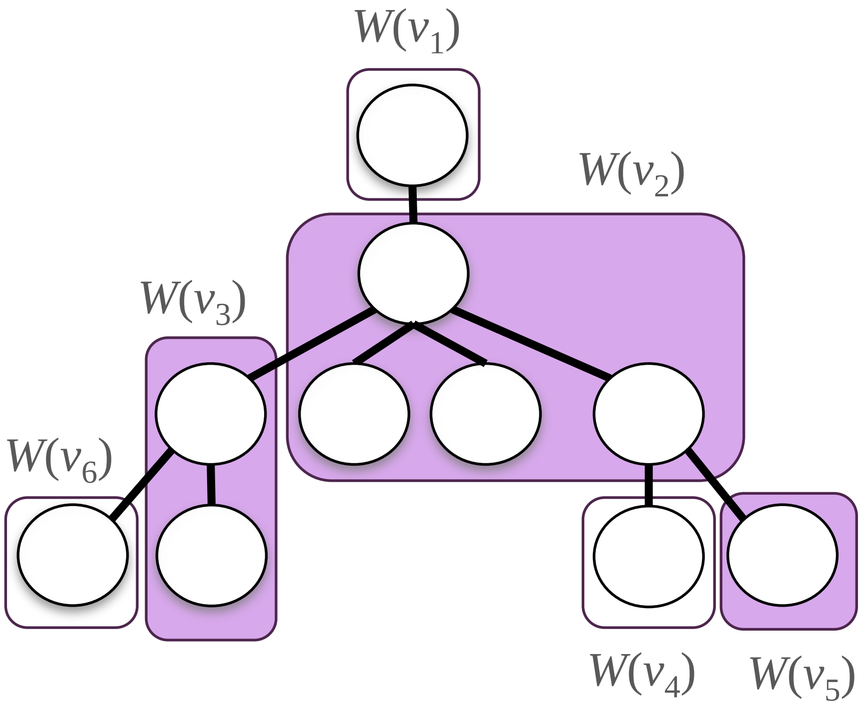
Tree contraction pattern (TC-pattern)

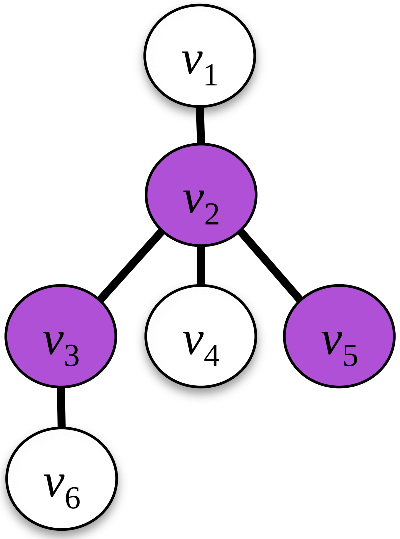
- A tree T=(V_T,E_T) with root r_T <u>*matches*</u> a TC-pattern t with root r_t, if there is a partition of V_T, {$W(v_1)$,…,$W(v_m)$} for $v_1,\ldots,v_m \in V_t$, such that
  1. for $v \in V_t \setminus U_t$, $W(v)$ includes exactly one vertex,
  2. for any $v \in V_t$, any pair of $W(v)$ is connectied,
  3. $W(r_t)$ includes r_T, and
  4. the tree obtained from T by merging all vertices in $W(v)$ into one vertex for each $v \in U_t$ is isomorphic to T.



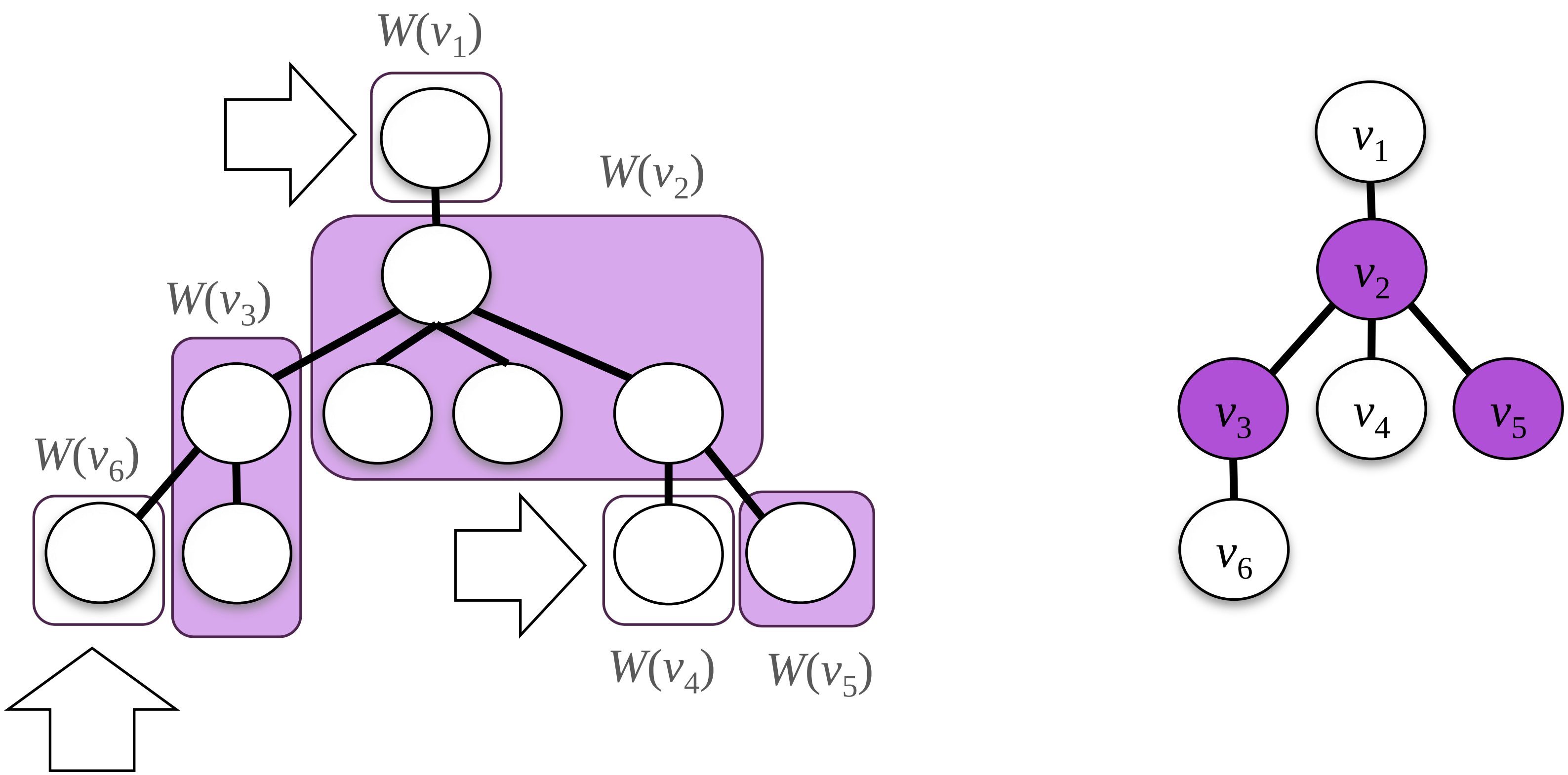


Tree contraction pattern (TC-pattern)



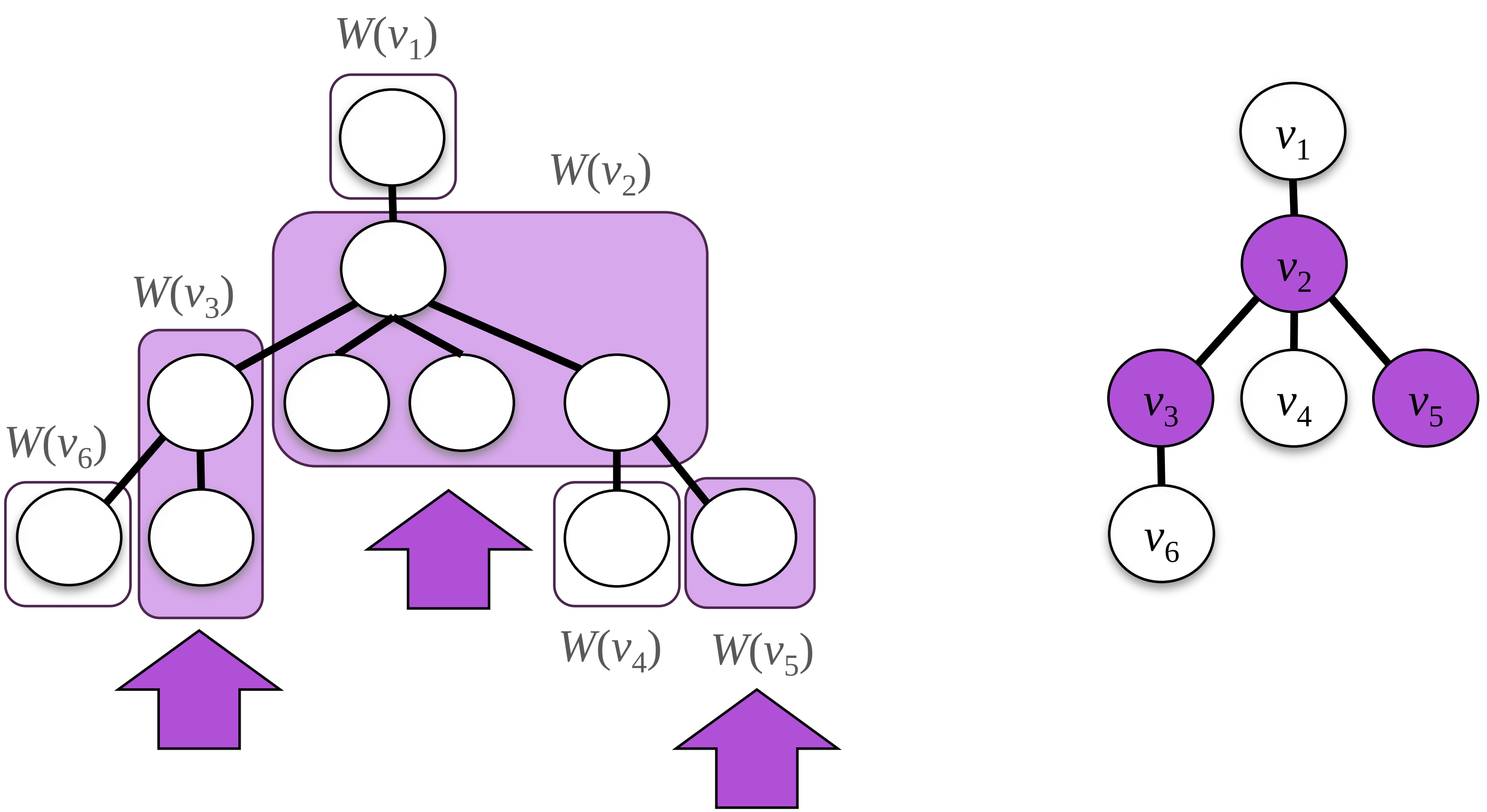


Tree contraction pattern (TC-pattern)



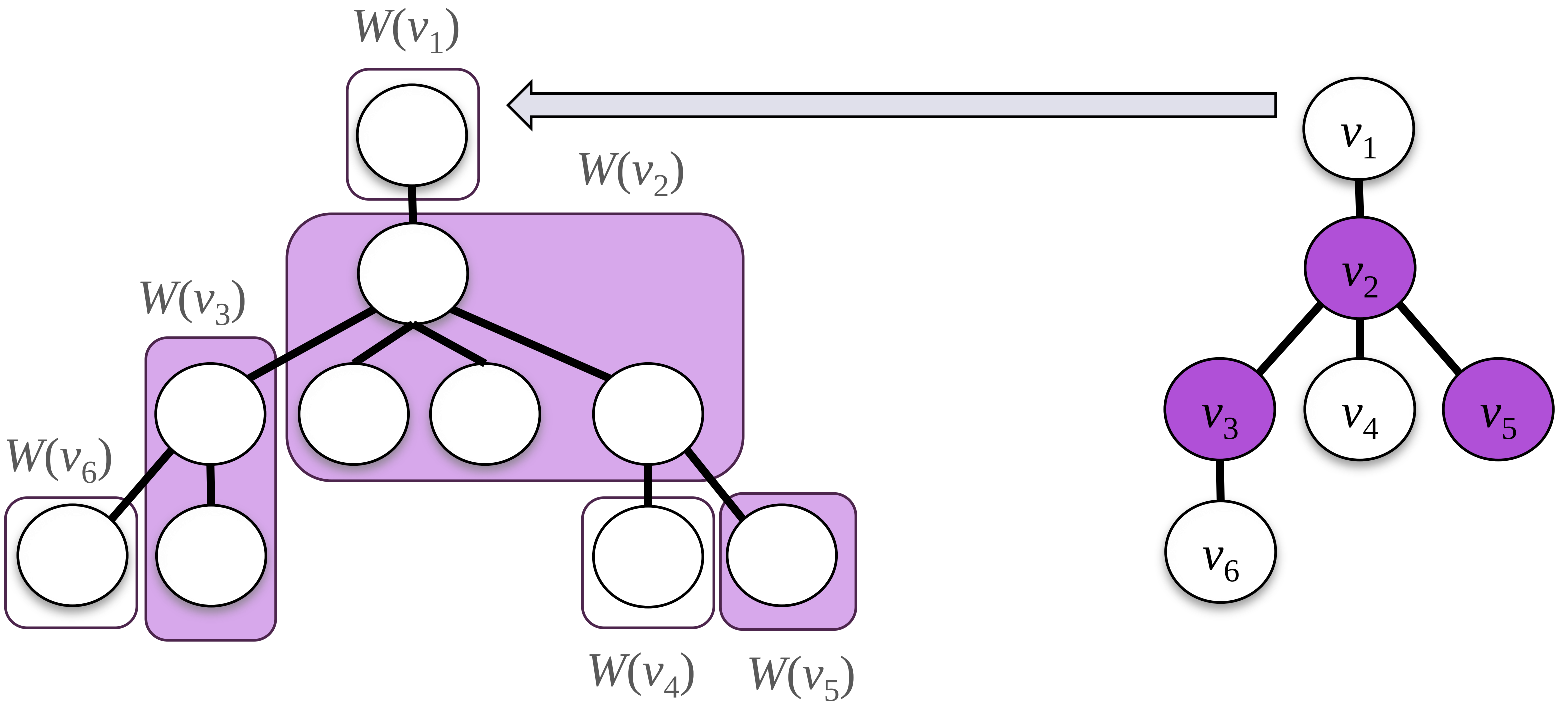
1. For $v \in V_t \setminus U_t$, $W(v)$ includes exactly one vertex.

Tree contraction pattern (TC-pattern)



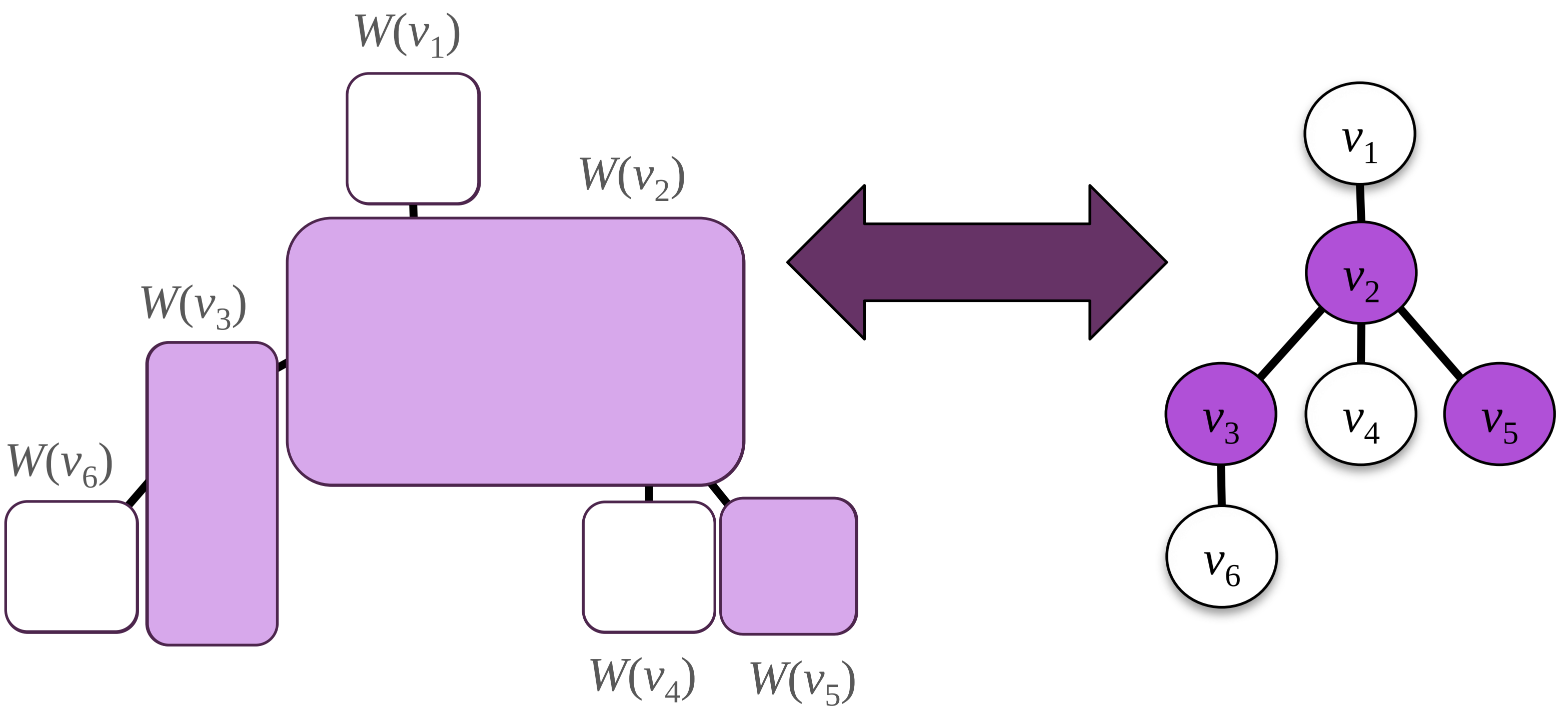
2. For any $v \in V_t$, the subtree induced by $W(v)$ is connected.

Tree contraction pattern (TC-pattern)



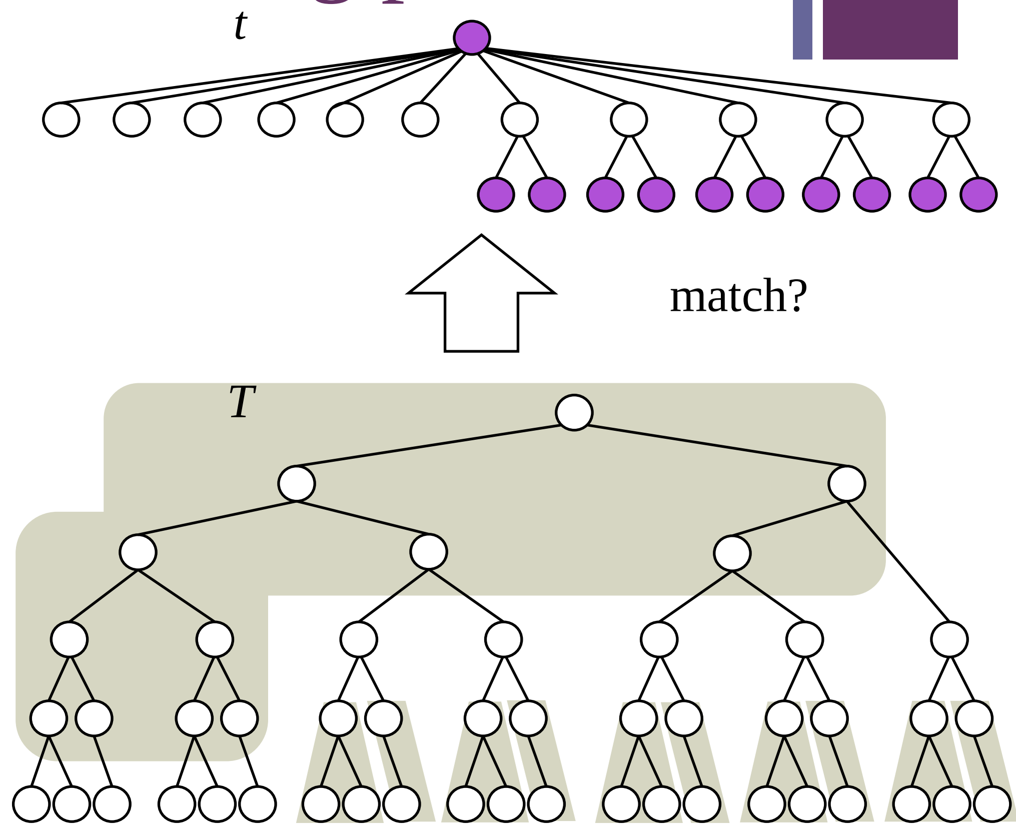
3. $W(r_t)$ includes r_T.

Tree contraction pattern (TC-pattern)



4. The tree obtained from T by merging all vertices in $W(v)$ into one vertex for each $v \in U_t$ is isomorphic to t.

Time complexity of the TC-pattern matching problem



TC-pattern matching problem

Input: a rooted unordered tree *T*, a TC-pattern *t*.

Question: *T* matches *t* ?

Theorem

TC-pattern matching problem is NP-complete.

Proof: Transform from X3C.

In this paper, we consider <u>a subclass of TC-patterns whose matching problem can be solved in polynomial time.</u>

Time complexity of the TC-pattern matching problem

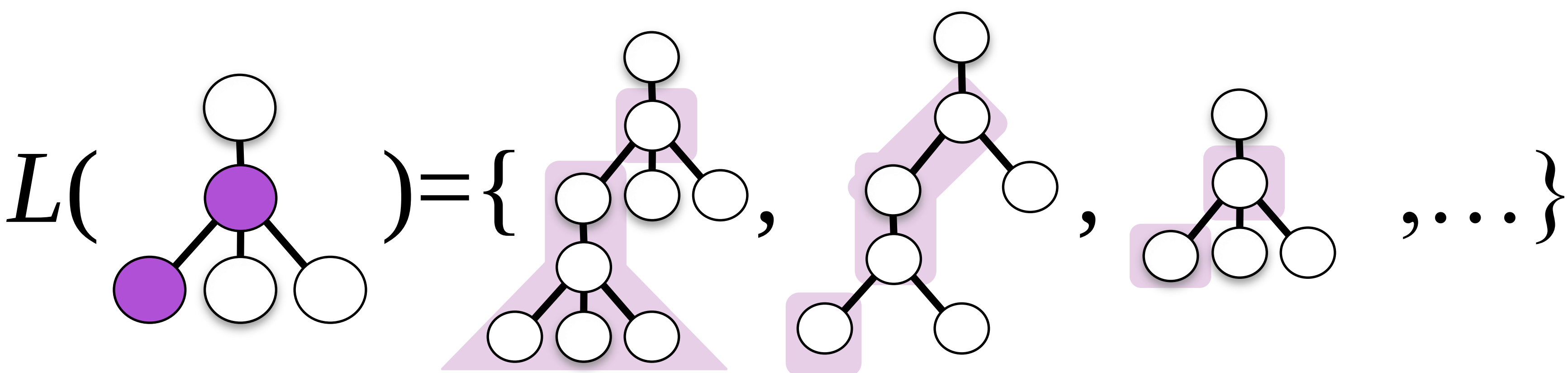
Theorem

We assume that the degree of every contractible vertex in TC-patterns is bounded by a constant d. Then TC-pattern matching problem for a given TC-pattern t and a given tree T is solved in $O(nN^{\max\{d-1,1.5\}})$ time, where $n=|V_t|$ and $N=|V_T|$.

Method: Dynamic Programing.
For every vertex v of T, we compute a unique label (a collection of subsets of t) by using the labels of the children of v.

The minimal language problem for TC-patterns

- The TC-pattern language $L(t)$
  - A representation power of TC-pattern t.

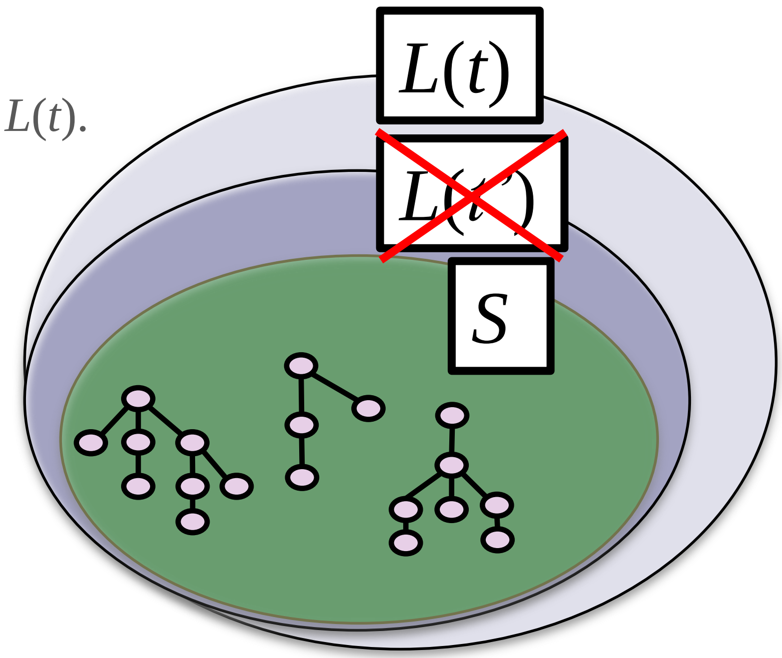


- The TC-pattern language $L(t)$ is defined as the set of all trees which match t.

The minimal language problem for TC-patterns

- *MINimal Language (MINL) problem* for TC-patterns
  - Instance: A set of rooted unordered trees $S=\{T_1,T_2,\ldots,T_m\}$
  - Problem: Find a minimally generalized TC-pattern explaining S.

- Def. *A minimally generalized TC-pattern t* explaining S
  1. $L(t)$ contains all trees in S.
  2. There is no TC-pattern t' such that $S \subseteq L(t') \subsetneq L(t)$.



- From a data mining point of view,
  this problem is a problem for searching a given
  dataset for only one specialized common pattern.

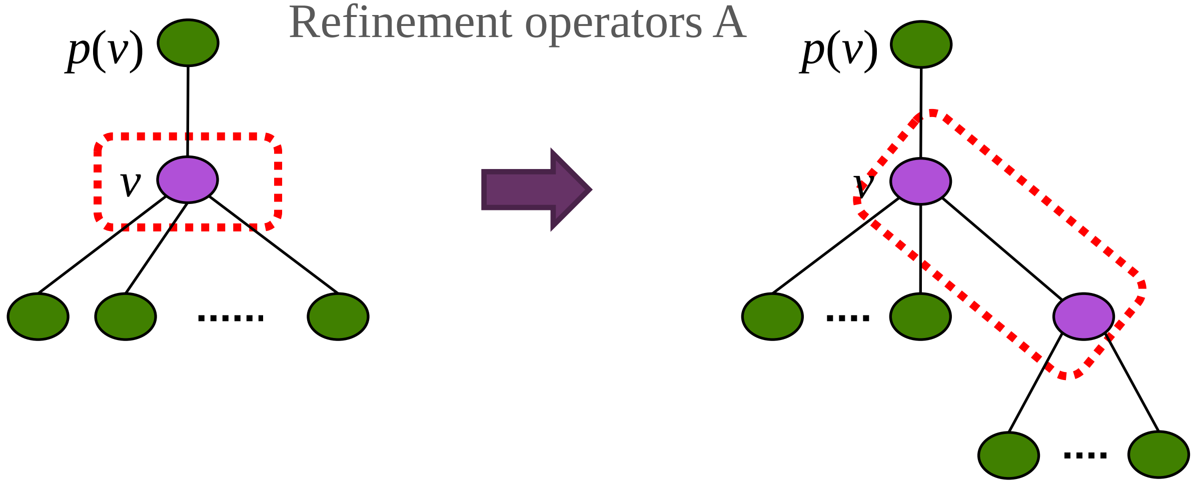
The minimal language problem for TC-patterns

- An idea to compute the MINL problem
  - Starting from the most generalized TC-pattern.
    - The most generalized TC-pattern is a TC-pattern consisting of only one contractible vertex.
  - Trying to specialize TC-pattern *t* to provide a more specialized TC-pattern *t'* which satisfies the next conditions.
    - $S \subseteq L(t') \subsetneq L(t)$, and
    - if there is no such *t'*, output *t*.

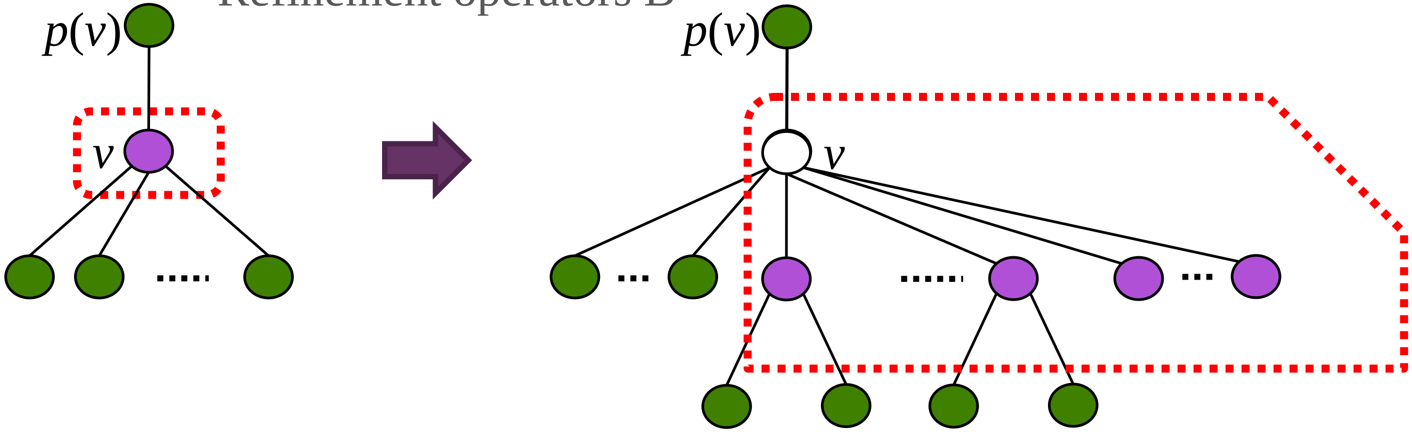
- Next, we show two refinement operators which are used in this refinement process.

The minimal language problem for TC-patterns

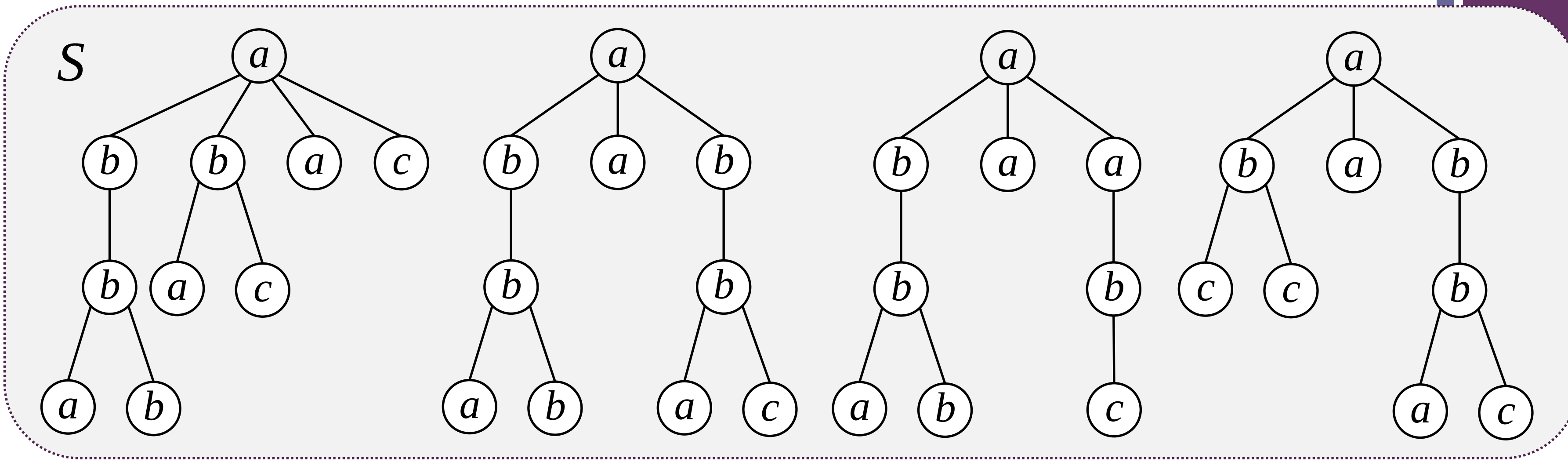
Refinement operators A

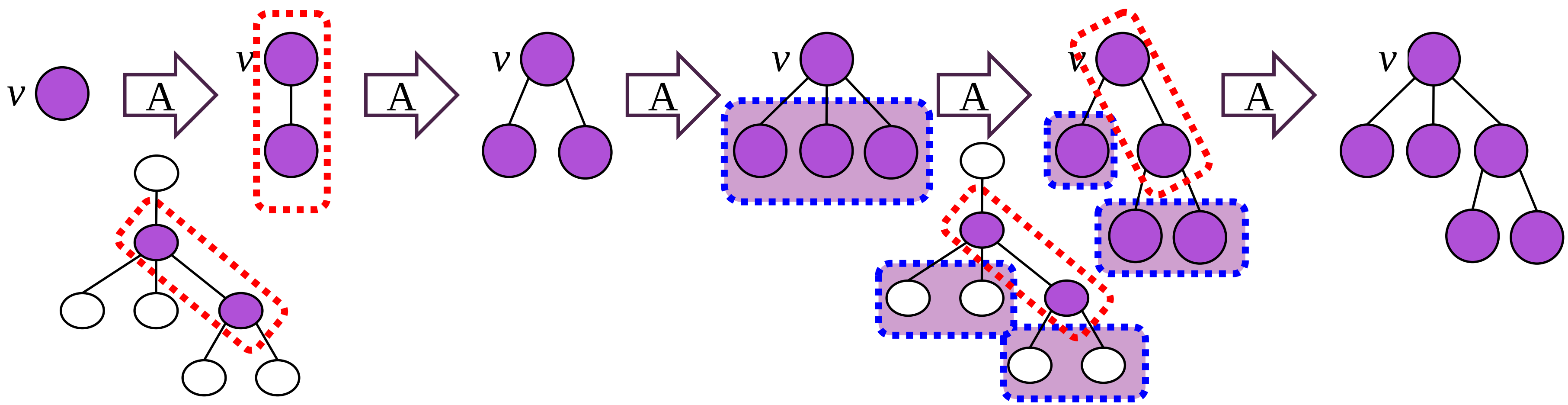


Refinement operators B

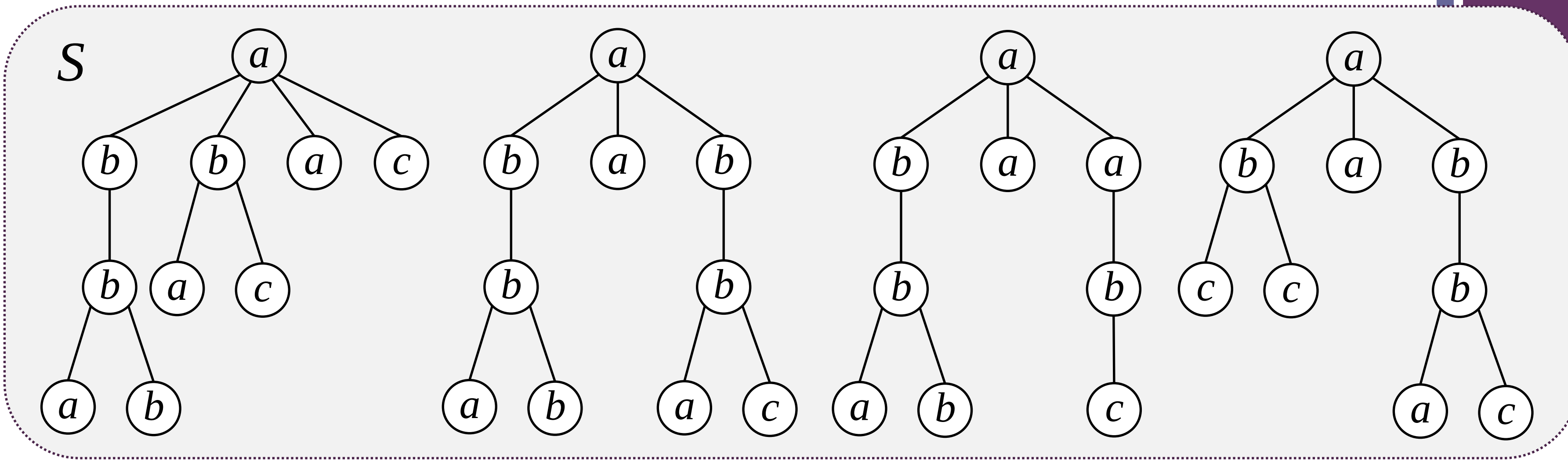


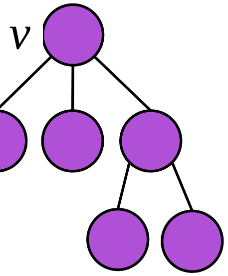
The minimal language problem for TC-patterns



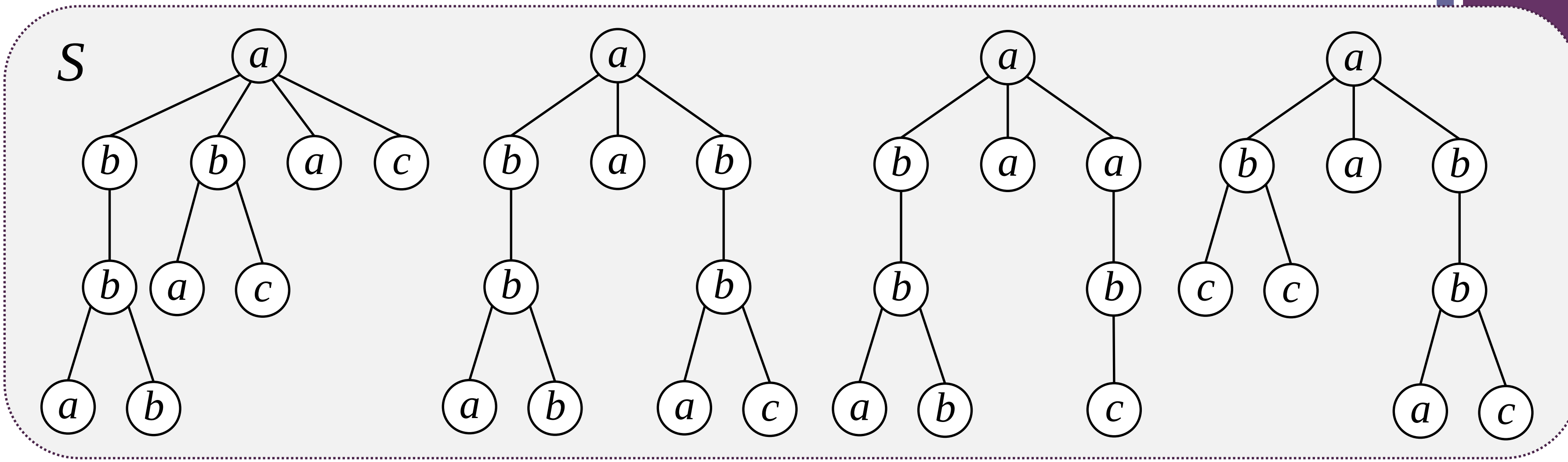


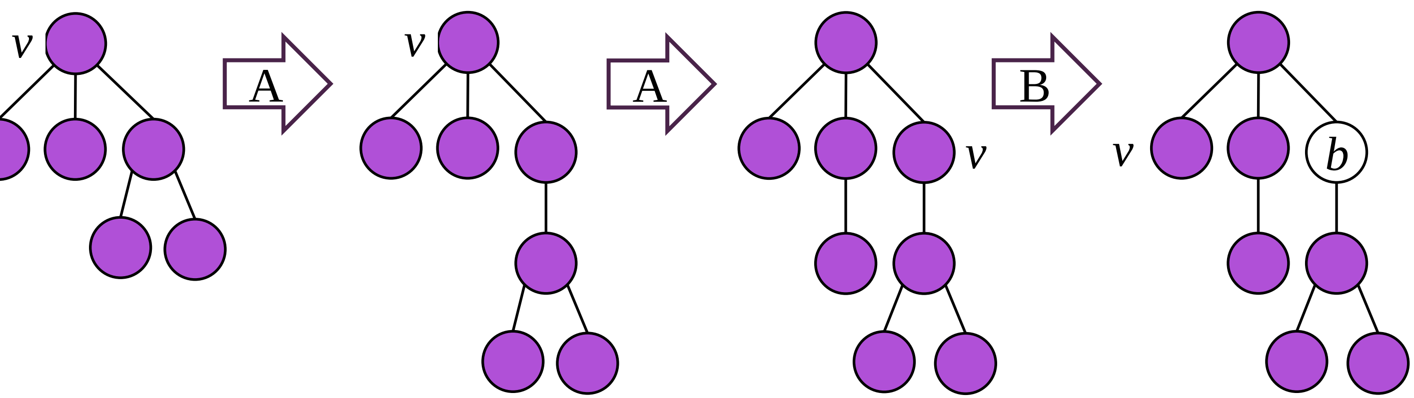
The minimal language problem for TC-patterns



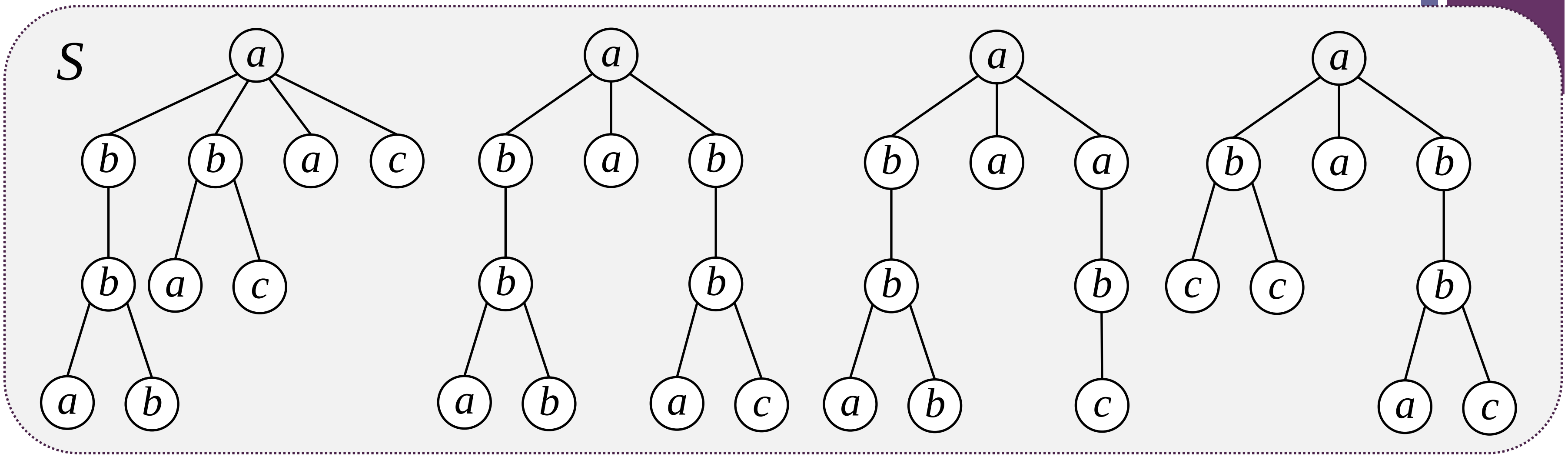


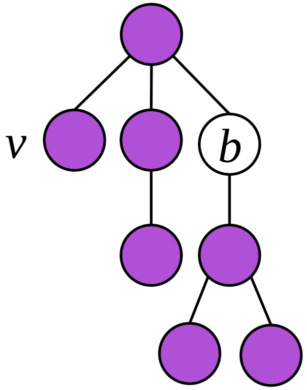
The minimal language problem for TC-patterns



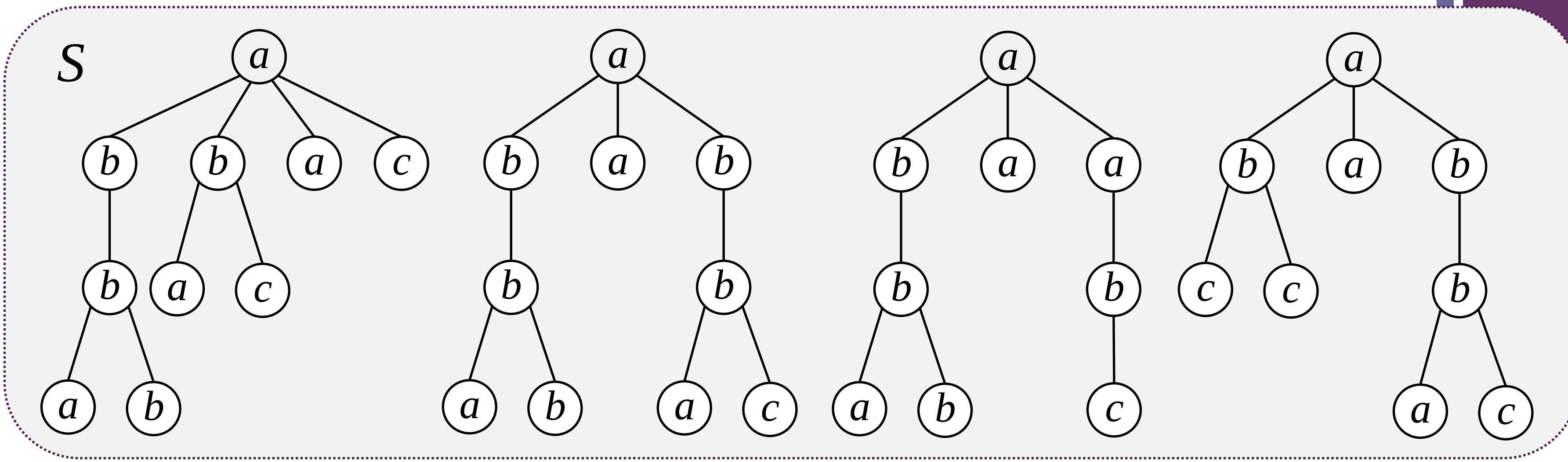


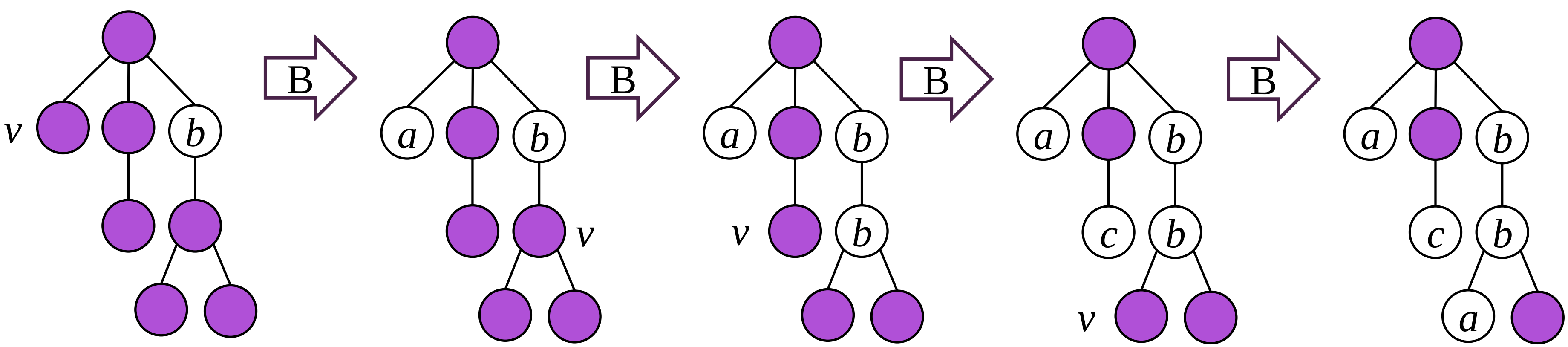
The minimal language problem for TC-patterns





The minimal language problem for TC-patterns





The minimal language problem for TC-patterns

- **Theorem**

We assume that there are infinitely many vertex labels in $\sum$, and that the degree of every contractible vertex in TC-patterns is bounded by constant d.

Minimal language problem for TC-patterns for a given set of trees S is computed in

$$O(nN_{\min}{}^{d+1}N_{\max}{}^{\max\{d\text{-}1,1.5\}}|S||\textstyle\sum(S)|) \text{ time,}$$

where $N_{\min}=\min_{T\in S}|V_T|$, $N_{\max}=\max_{T\in S}|V_T|$, and $\sum(S)=\{\delta\in\sum \mid \delta$ appears in $S\}$.

Conclusions

- A learning model on computational learning theory: A polynomial time inductive inference from a positive data
- **Theorem**[Angluin, ’80, Shinohara, ’82]: If a class C has finite thickness, and the membership and the minimal language (MINL) problems for C are solvable in polynomial time, class C is polynomial time inductively inferable from positive data.
- **Corollary**

  The class of TC-patterns such that <u>the degree of every contractible vertex in it is bounded by a constant *d*</u> is polynomial time inductively inferable from positive data.

Future work

- Experiment of our algorithm
  - Web document
  - Sugar chain data etc..
- Development of more fast algorithm to find a minimally generalized TC-pattern.
- To consider graph contraction patterns (GC-patterns) based on tree contraction patterns (TC-patterns) .
  - outerplanar graph
  - bounded treewidth graph etc..