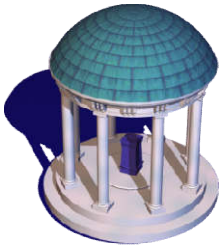


Mining Frequent Subgraphs

COMP 790-90 Seminar

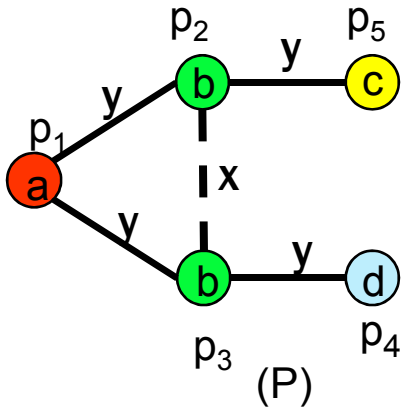
Spring 2011

FFSM: Fast Frequent Subgraph Mining -- An Overview:

- How to solve graph isomorphism problem?
  - A Novel Graph Canonical Form: CAM
- How to tackle subgraph isomorphism problem (NP-complete)?
  - Incrementally maintained embeddings
- How to enumerate subgraphs:
  - An Efficient Data Structure: CAM Tree
  - Two Operations: CAM-join, CAM-extension.

Adjacency Matrix

- Every diagonal entry of adjacency matrix M corresponds to a distinct vertex in G and is filled with the label of this vertex.
- Every off-diagonal entry in the lower triangle part of M[1] corresponds to a pair of vertices in G and is filled with the label of the edge between the two vertices and zero if there is no edge.



a				
y	b			
y	x	b		
0	y	0	c	
0	0	y	0	d

M_1

a				
y	b			
y	x	b		
0	0	y	d	
0	y	0	0	c

M_2

b				
x	b			
y	0	d		
0	y	0	c	
y	y	0	0	a

M_3

[1]for an undirected graph, the upper triangle is always a mirror of the lower triangle
Throughout this paper, we assume the following **total order $a \geq b \geq x \geq y \geq 0$**

Code

- A *Code* of n × n adjacency matrix *M* is defined as sequence of lower triangular entries (including the diagonal entries) in the order:

  $M_{1,1}\ M_{2,1}\ M_{2,2}\ \ldots\ M_{n,1}\ M_{n,2}\ \ldots M_{n,n-1}\ M_{n,n}$

a				
y	b			
y	x	b		
0	y	0	c	
0	0	y	0	d

M_1

a				
y	b			
y	x	b		
0	0	y	d	
0	y	0	0	c

M_2

b				
x	b			
y	0	d		
0	y	0	c	
y	y	0	0	a

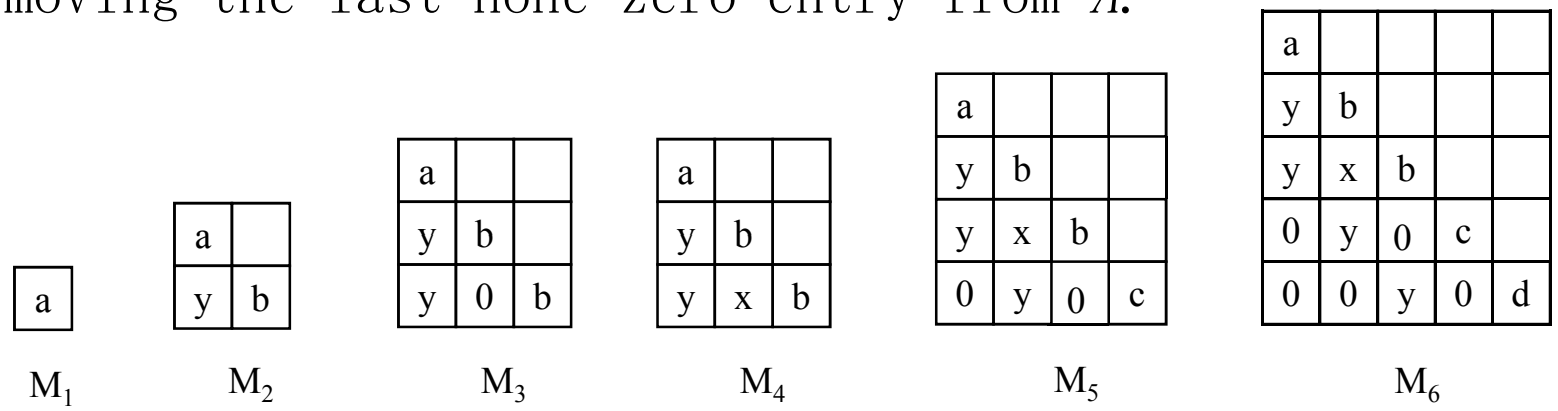
M_3

Code(M_1): aybyxb0y0c00y0d
Code(M_2): aybyxb00yd0y00c
Code(M_3): bxby0d0y0cyy00a

- The *Canonical Adjacency Matrix* is the one produces the **maximal** code, using lexicographic order.

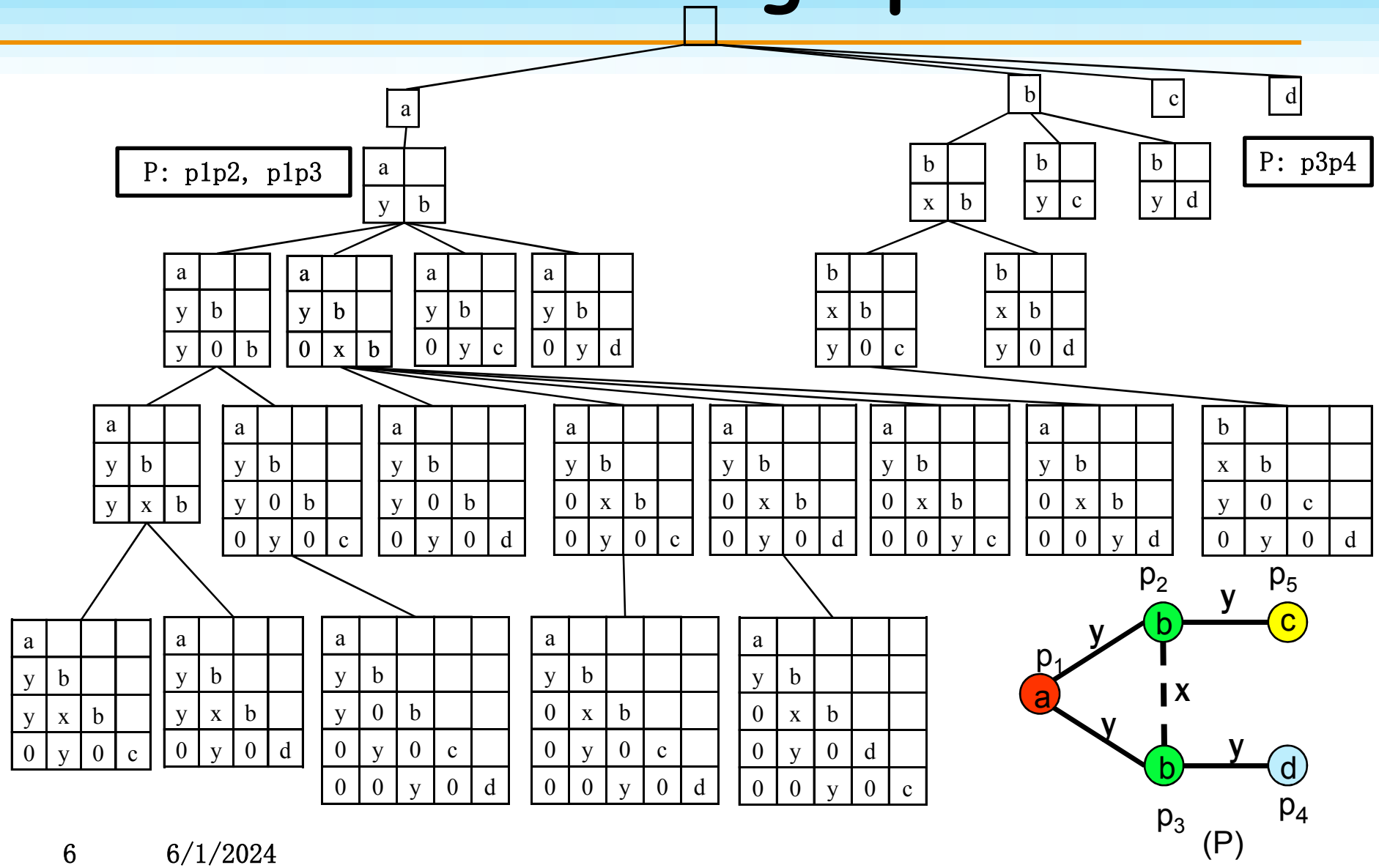
MP Submatrix

- For an $m \times m$ matrix A, an $n \times n$ *matrix* B is A's *maximal proper submatrix* (MP Submatrix), iff B is obtained by removing the last none-zero entry from A.

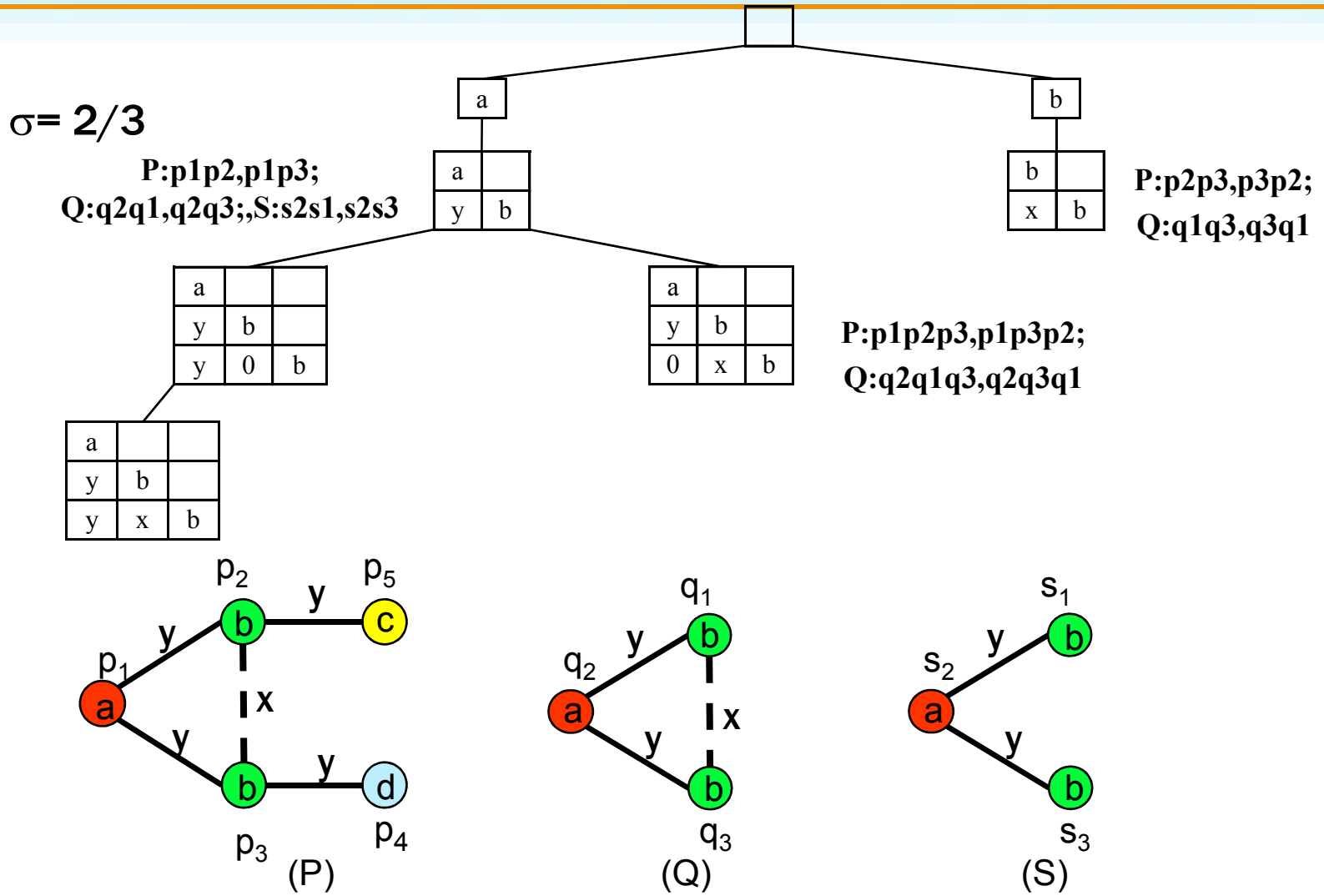


  - We define a CAM is *connected* iff the corresponding graph is connected.
  - Theorem I: A CAM's MP submatrix is CAM
  - Theorem II: A connected CAM's MP submatrix is connected

CAM Tree: Subgraphs



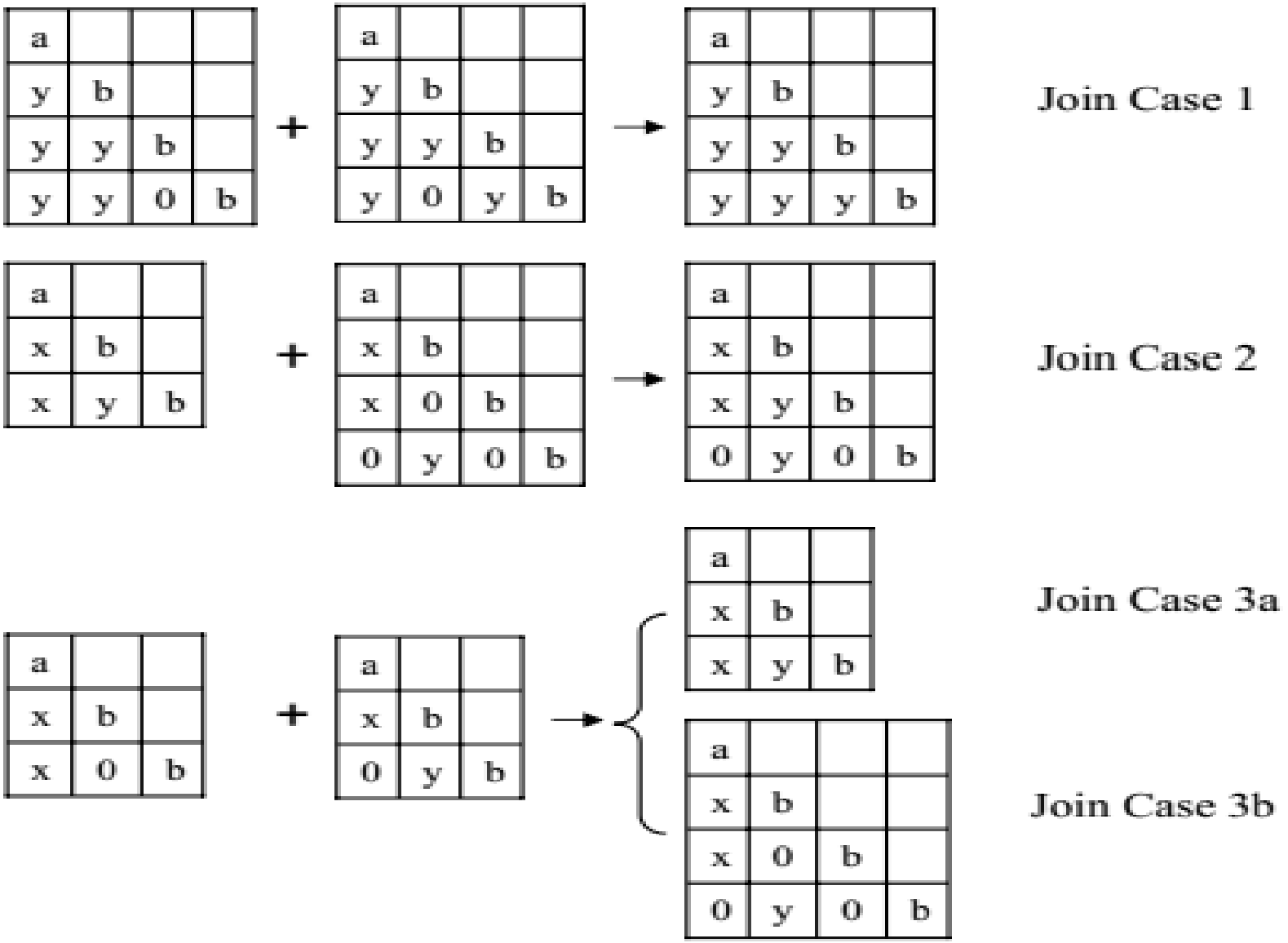
CAM Tree: Frequent Subgraphs



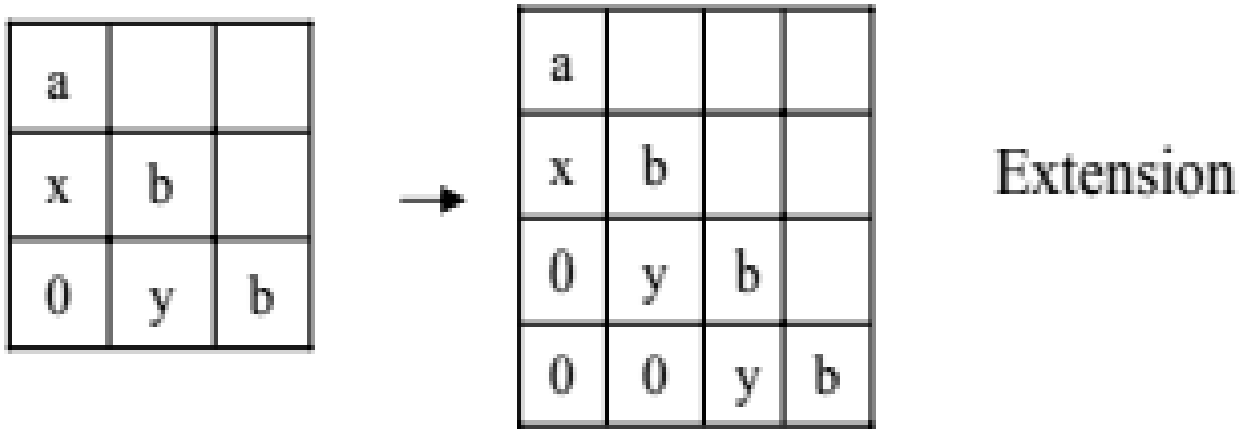
How to Enumerate Nodes in a CAM Tree?

- Two operations to explore CAM tree:
  - CAM-Join
  - CAM-Extension
- Augmenting CAM tree with Suboptimal CAMs
- Objectives:
  - no false dismissal
  - no redundancy
- Plus: We want to this efficiently!

Examples of the join operation

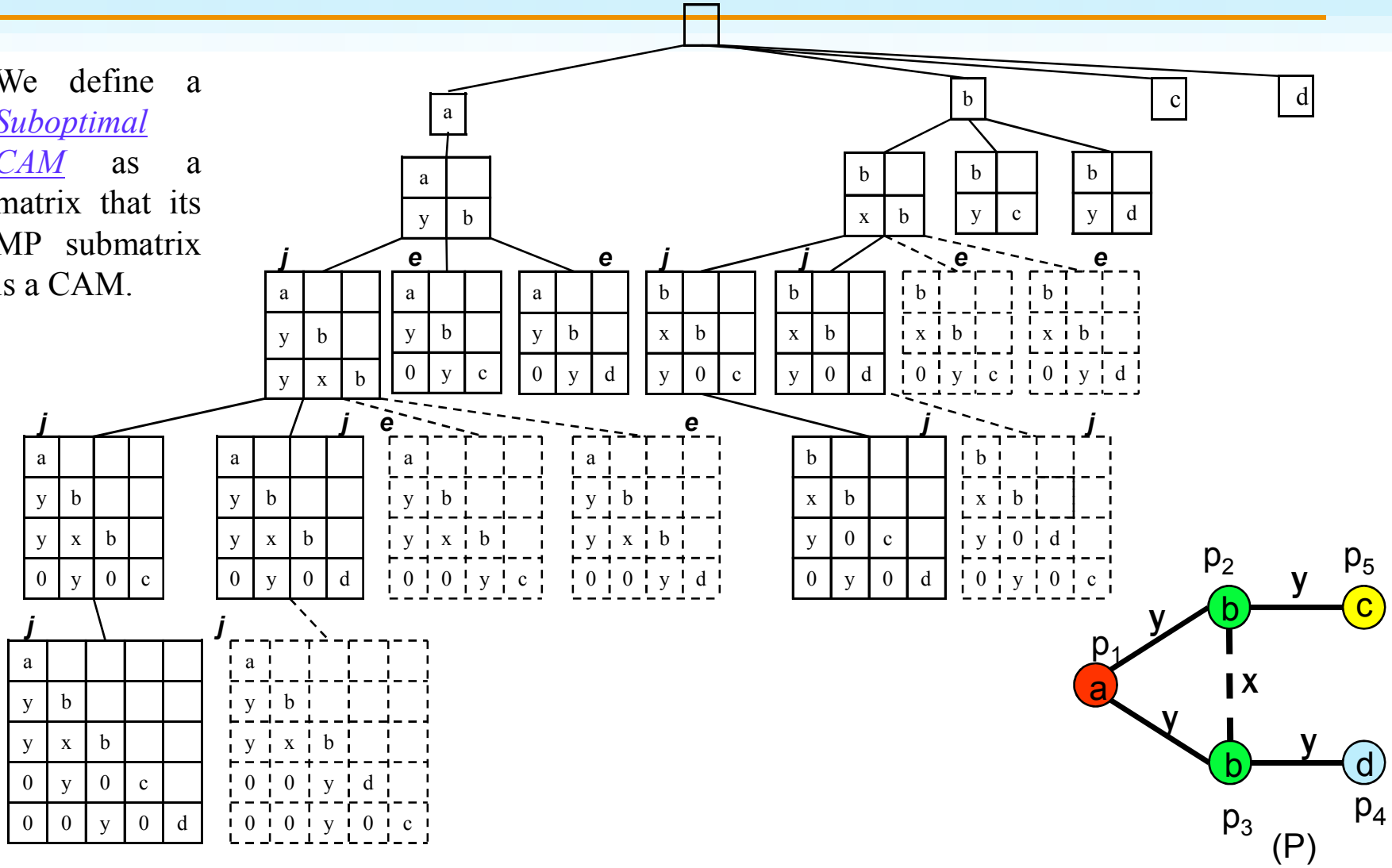


Examples of the extension operation



Suboptimal Tree

We define a *Suboptimal CAM* as a matrix that its MP submatrix is a CAM.



Performance Enhancement using an Embedding List

Definition 3.3 *Given an arbitrary $n \times n$ adjacency matrix A and a labeled graph $G = (V, E, \Sigma_V, \Sigma_E, l)$, a vertex list $L = u_1, u_2, \ldots, u_n \subset V$ is an* ***embedding*** *of A in G iff:*

$$(i) \quad \forall\, i, (a_{i,i} = l(u_i));$$

$$(ii) \quad \forall\, i, j (a_{i,j} \neq 0 \Rightarrow a_{i,j} = l(u_i, u_j));$$

where $0 < j < i \leq n$.

Performance Enhancement using an Embedding List

From the above analysis, we conclude that for the embedding set O_A of a suboptimal CAM A, which is joined by two suboptimal CAMs P and Q through join case 1, we have $O_A = O_P \cap O_Q$, where O_P and O_Q are the embedding sets of suboptimal CAM P and Q, respectively.

Summary

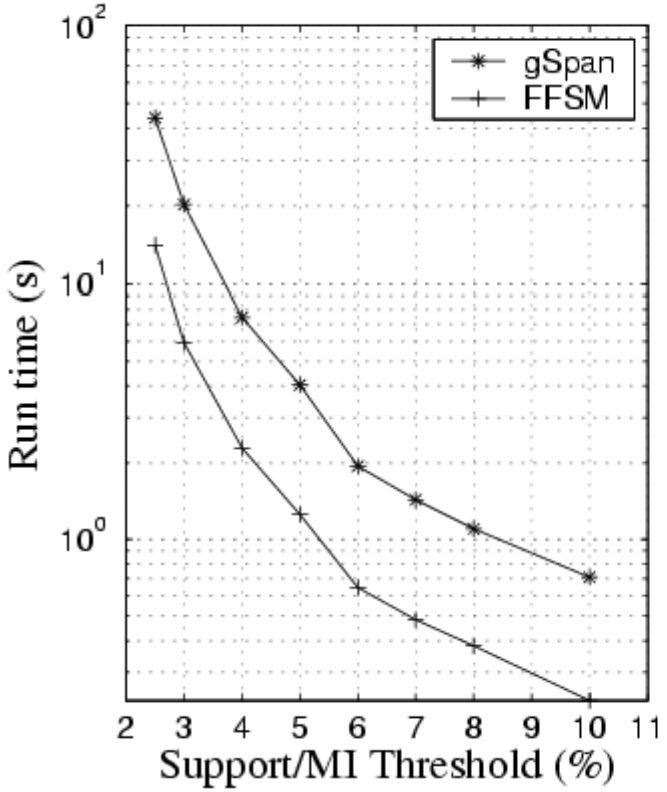
- Theorem:

  For a graph G, let C_{K-1} (C_k) be set of the suboptimal CAMs of all size-(K-1) (K) subgraphs of G ($K \geq 2$). Every member of set C_K can be enumerated unambiguously either by ***joining*** two members of set C_{K-1} or by ***extending*** a member in C_{K-1}.

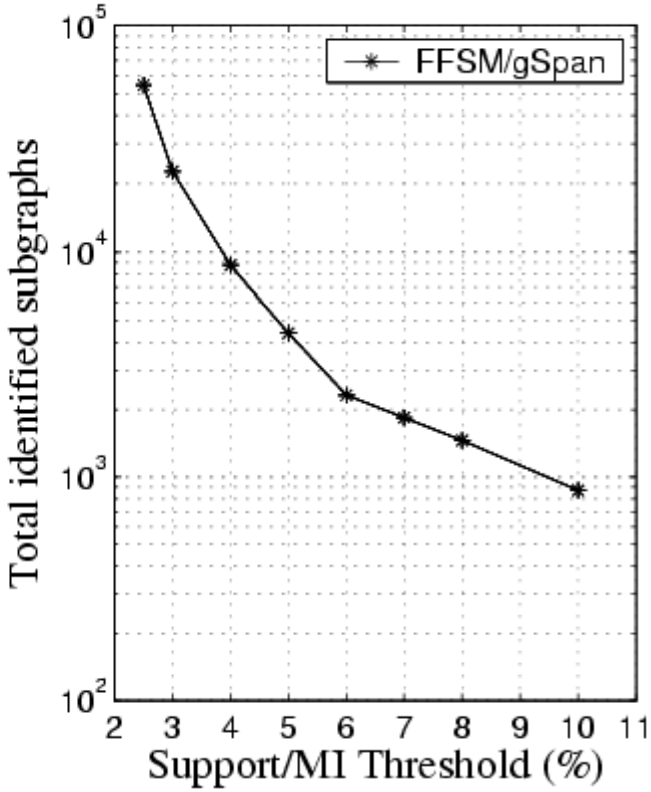
Experimental Study

- Predictive Toxicology Evaluation Competition (PTE)
  - Contains: 337 compounds
  - Each graph contains 27 nodes and 27 edges on average
- NIH DTP Anti-Viral Screen Test (DTP CA/CM)
  - Chemicals are classified to be Confirmed Active (CA), Confirmed Moderate Active (CM) and Confirmed Inactive (CI).
  - We formed a dataset contains CA (423) and CM (1083).
  - Each graph contains 25 nodes and 27 edges on average

Performance (PTE)



Support Threshold (%)



Support Threshold (%)

Performance (DTP CACM)