# Mining Frequent Patterns in Print Logs with Semantically Alternative Labels

Xin Li

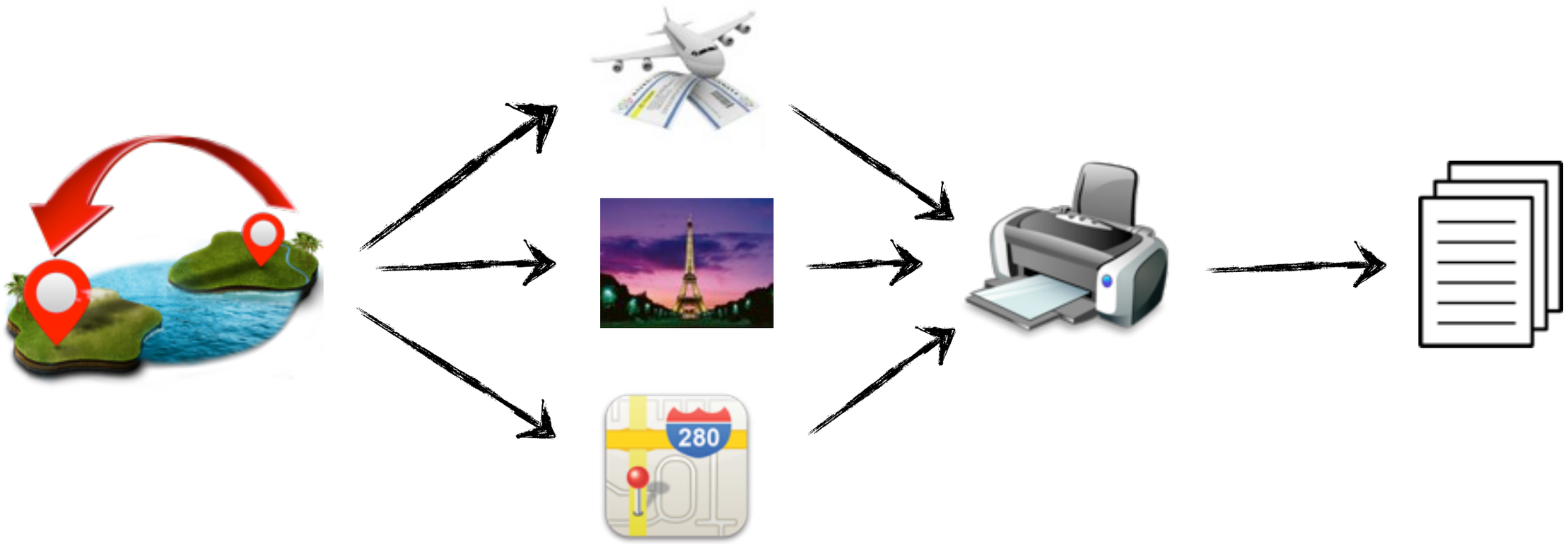
Sept. 4, 2013

# On the menu

- Background
- Problem Statement
- Algorithm(Basic & PaSAL)
- Experiment
- Conclusion

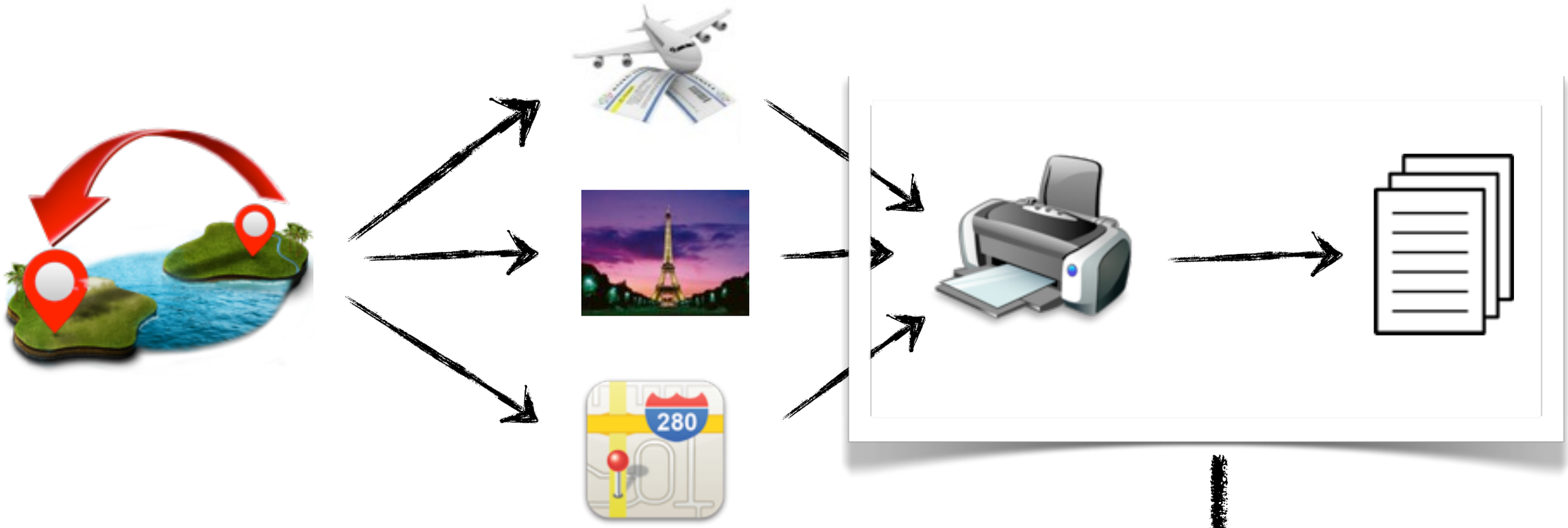
# Background 1: Print Logs

## Scenario 1: Travel



# Background 1: Print Logs

## Scenario 1: Travel



# Background 1: Print Logs

## Scenario 2: Printing Tools


Before

After


Print Logs

# Background 1: Print Logs

Print Logs: logs collected from the printer

| User | Printed URLs |
|---|---|
| $user_1$ | http://maps.google.com/maps?saddr=27400+Old+Trilby+Roa |
| | http://www.groupon.com/deals?city=houston |
| | http://www.booking.com/hotel/us/the-huston.en.html |
| $user_2$ | ..... |
| | http://maps.google.com/maps?saddr=2800+League+City+Parkwa |
| ..... | ..... |

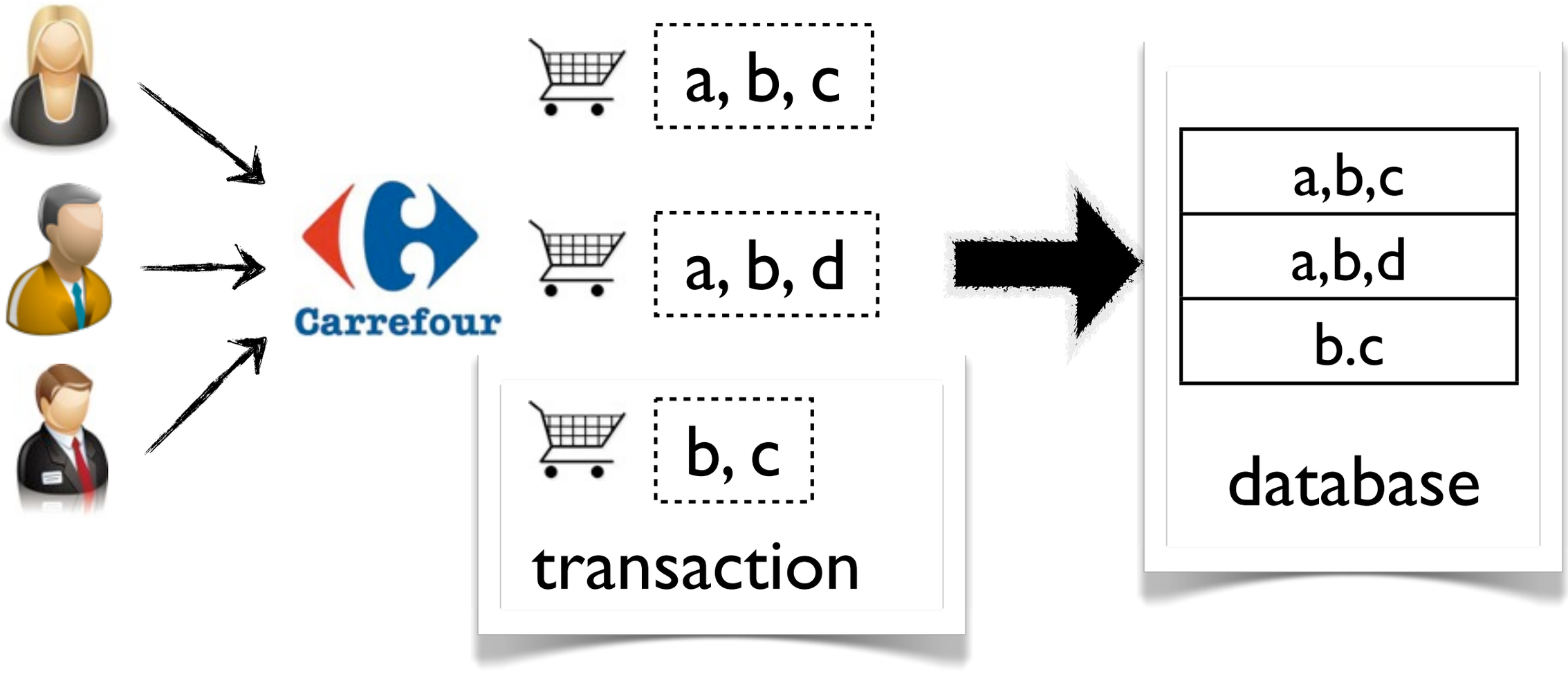
Goal: capture users' intention

Applications: printing recommendation

behavior targeting

# Background 2: Pattern Mining

## Scenario: shopping cart



# Background 2: Pattern Mining

| tid | Itemset |
|---|---|
| 1 | a,b,c |
| 2 | a,b,d |
| 3 | b,c |
| 4 | a,c |
| 5 | b,c,d |

database

$I = \{a, b, c, d\} \quad T = (tid, X), X \subseteq I$

frequency(X): number of transactions that contain X

$frequency(a) = 3$

support(X): frequency(X)/total number of transaction

$support(a) = 3/5 = 0.6$

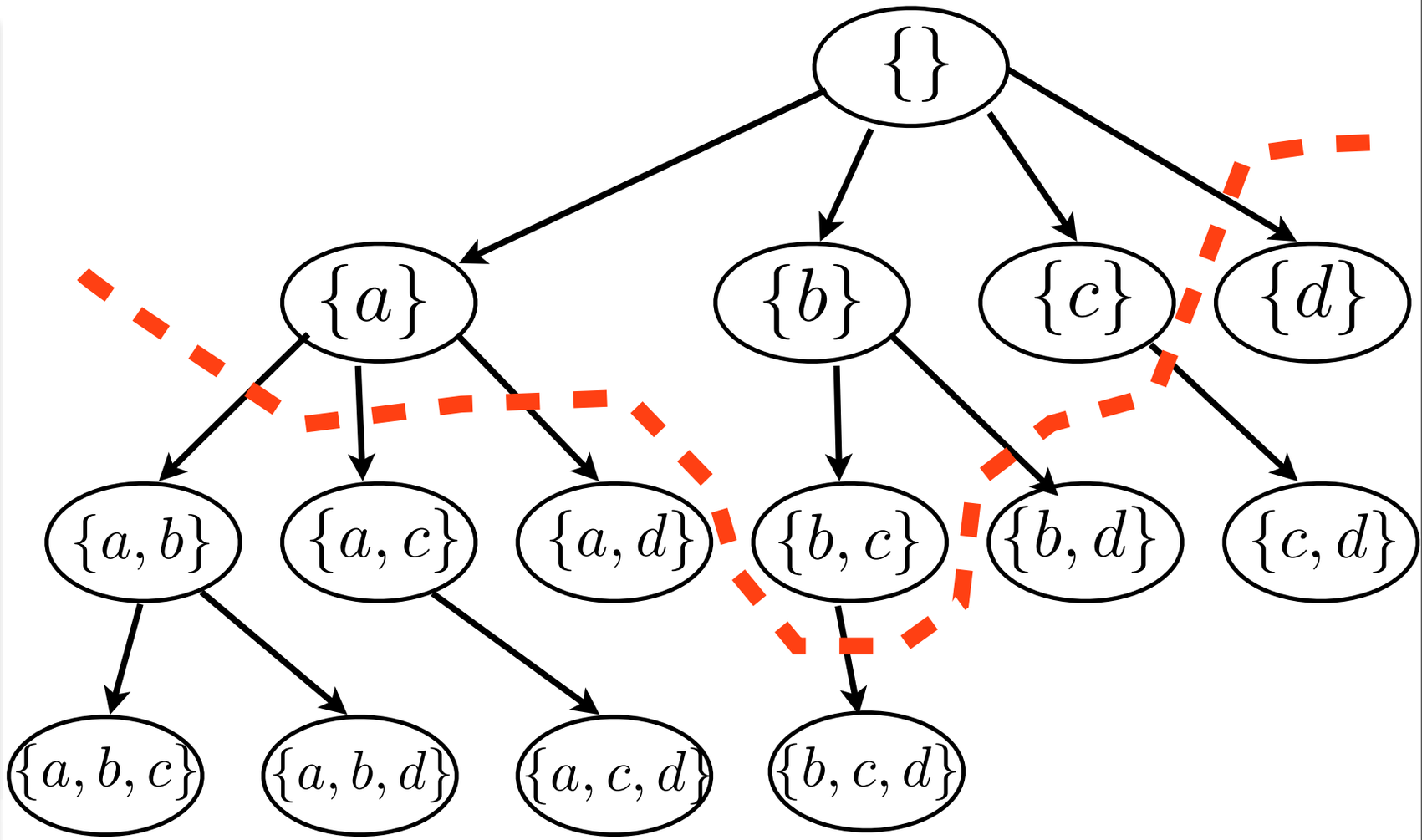
frequent pattern X: support(X)>=threshold

Set threshold =0.6 $\{a\}, \{b\}, \{c\}, \{b, c\}$

# Background 2: Pattern Mining

| tid | Itemset |
|---|---|
| 1 | a,b,c |
| 2 | a,b,d |
| 3 | b,c |
| 4 | a,c |
| 5 | b,c,d |

database




support: anti-monotonicity

property with anti-monotonicity can be used for pruning

Algorithms: Apriori, Eclat, FP-growth, ...

# Problem Statement

| tid | Itemset |
|---|---|
| 1 | a,b,c |
| 2 | a,b,d |
| 3 | b,c |
| 4 | a,c |
| 5 | b,c,d |

database

| uid | URLset |
|---|---|
| 1 | u1,u2,u3 |
| 2 | u1,u2,u4 |
| 3 | u2,u3 |
| 4 | u1,u3 |
| 5 | u2,u3,u4 |

database

| User | Printed URLs |
|---|---|
| $user_1$ | http://maps.google.com/maps?saddr=27400+Old+Trilby+Roa |
| | http://www.groupon.com/deals?city=houston |
| | http://www.booking.com/hotel/us/the-huston.en.html |
| $user_2$ | ..... |
| | http://maps.google.com/maps?saddr=2800+League+City+Parkwa |
| ..... | ..... |

# Problem Statement

mining frequent patterns in print logs

| tid | Itemset |
|---|---|
| 1 | a,b,c |
| 2 | a,b,d |
| 3 | b,c |
| 4 | a,c |
| 5 | b,c,d |

database

| uid | URLset |
|---|---|
| 1 | u1,u2,u3 |
| 2 | u1,u2,u4 |
| 3 | u2,u3 |
| 4 | u1,u3 |
| 5 | u2,u3,u4 |

database

| User | Printed URLs |
|---|---|
| $user_1$ | http://maps.google.com/maps?saddr=27400+Old+Trilby+Roa |
| | http://www.groupon.com/deals?city=houston |
| | http://www.booking.com/hotel/us/the-huston.en.html |
| $user_2$ | ..... |
| | http://maps.google.com/maps?saddr=2800+League+City+Parkwa |
| ..... | ..... |

# Problem Statement

| User | Printed URLs |
|---|---|
| $user_1$ | http://maps.google.com/maps?saddr=27400+Old+Trilby+Roa |
| | http://www.groupon.com/deals?city=houston |
| | http://www.booking.com/hotel/us/the-huston.en.html |
| $user_2$ | ..... |
| | http://maps.google.com/maps?saddr=2800+League+City+Parkwa |
| ..... | ..... |

google map

data sparsity & pattern interpretability

$http://maps.google.com/maps?saddr = 27400 + Old + Trilby + Roa$

$http://maps.google.com/maps$

Semantically Alternative Labels

# Problem Statement

Assumption: Printing URLs with the same domain have the same intention

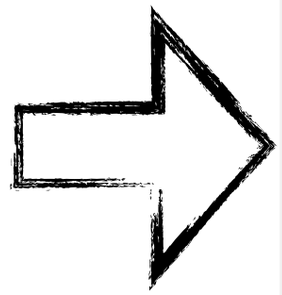
URL Domain → delicious.com → Top 10 tags

| Domain | Tag Representations |
|---|---|
| maps.google.com | maps, google, travel, map, reference, search, directions, tools, clear-lake, brian |
| www.groupon.com | shopping, coupons, deals, social, discount, coupon, business, crowd-sourcing, marketing, travel |
| www.booking.com | travel, hotel, hotels, booking, accommodation, search, viajes, online, hoteles, turismo |

# Problem Statement

| uid | URLset |
|---|---|
| 1 | u1,u2,u3 |
| 2 | u1,u2,u4 |
| 3 | u2,u3 |
| 4 | u1,u3 |
| 5 | u2,u3,u4 |

database



| uid | Tagset |
|---|---|
| 1 | t1,t2, ... , tN |
| 2 | t2, t3, ... , tK |
| 3 | t2, t4, ... , tM |
| 4 | ... |
| 5 | ... |

database

Annoy: generating large numbers of patterns

# Problem Statement

## Annoy: generating large numbers of patterns

| Domain | Tag Representations |
|---|---|
| maps.google.com | maps, google, travel, map, reference, search, directions, tools, clear-lake, brian |

$\{maps, google\}$ $\{maps, travel\}$ $\{maps, travel, search\}$ $2^{10}$

cross-domain pattern: none of the tags in the patterns come from the same domain

Restrict Condition

PaSAL: frequent Patterns with Semantically Alternative Labels

Conflict Matrix

# Problem Statement

## Restrict Condition

| Domain | Tag Representations |
|---|---|
| maps.google.com | maps, google, travel, map, reference, search, directions, tools, clear-lake, brian |
| www.groupon.com | shopping, coupons, deals, social, discount, coupon, business, crowd-sourcing, marketing, travel |
| www.booking.com | travel, hotel, hotels, booking, accommodation, search, viajes, online, hoteles, turismo |

## Conflict Matrix

| Domain | Tag Representations |
|---|---|
| google | map, travel, search |
| groupon | coupon,travel |
| booking | travel, search, hotel |

$$\begin{array}{c} \\ maps \\ travel \\ search \\ hotel \\ coupon \end{array} \begin{array}{c} \begin{array}{ccc} google & groupon & booking \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

stored in bitmap

# Problem Statement

(a) Sampled Database $\mathcal{T}^{trans}$

| Transaction Id | Tags |
|---|---|
| 100 | maps, hotel, search |
| 200 | maps, hotel, coupon |
| 300 | travel, search, hotel, maps |
| 400 | coupon, search |
| 500 | search, travel, coupon |
| 600 | travel, search |

(b) Frequent Patterns

| Pattern | Support |
|---|---|
| ✓maps, hotel | 2 |
| ✓coupon, search | 2 |
| maps, search | 2 |
| travel, search | 3 |
| hotel, search | 2 |

$$
\begin{array}{c|ccc}
 & google & groupon & booking \\
maps & 1 & 0 & 0 \\
travel & 1 & 1 & 1 \\
search & 1 & 0 & 1 \\
hotel & 0 & 0 & 1 \\
coupon & 0 & 1 & 0
\end{array}
$$

$$
\begin{array}{rl}
maps = \{1,0,0\} & \\
search = \{1,0,1\} & \text{AND} \\
\hline
\{maps, search\} & \{1,0,0\} \\
\\
maps = \{1,0,0\} & \\
hotel = \{0,0,1\} & \text{AND} \\
\hline
\{maps, search\} & \{0,0,0\}
\end{array}
$$

# Problem Statement

| Transaction Id | Tags |
|---|---|
| 100 | maps, hotel, search |
| 200 | maps, hotel, coupon |
| 300 | travel, search, hotel, maps |
| 400 | coupon, search |
| 500 | search, travel, coupon |
| 600 | travel, search |

Given Database

$$\begin{array}{c} \\ maps \\ travel \\ search \\ hotel \\ coupon \end{array} \begin{array}{c} \begin{array}{ccc} google & groupon & booking \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

And CM(Conflict Matrix)/
RC(Restrict Condition)

| Pattern | Support |
|---|---|
| ✓maps, hotel | 2 |
| ✓coupon, search | 2 |
| maps, search | 2 |
| travel, search | 3 |
| hotel, search | 2 |

Mining frequent patterns that tags in RC do not co-occur in result patterns

# Algorithm Basic

| Pattern | Support |
|---|---|
| ✓maps, hotel | 2 |
| ✓coupon, search | 2 |
| maps, search | 2 |
| travel, search | 3 |
| hotel, search | 2 |

**post-process:**

step 1: do a normal frequent pattern mining algorithm

step 2: remove unqualified patterns

**Algorithm 1:** BASIC

**input** : the tag transaction data $\mathcal{T}^{trans}$, lexicographic ordering $\mathcal{T}$, the conflict matrix $CM$
**output**: restrict condition constrained patterns $\mathcal{P}$

```
1  Root.head ← empty;
2  Root.tail ← T;
3  DFS(Root);
4  DFS(Node) begin
5      for tag ∈ Node.tail do
6          p_tmp ← Node.head ∪ tag;
7          if p_tmp.frequency > min_sup × |T^trans| then
8              Node_tmp.head ← p_tmp;
9              remove tag from Node_tmp.tail ← Node.tail;
10             Children ← Node_tmp;
11             P_candidate ← Node_tmp.head;
12     for node ∈ Children do
13         DFS(node);
14 for pattern ∈ P_candiate do
15     if pattern violate CM then
16         remove pattern from P_candidate;
17 P ← P_candidate;
```

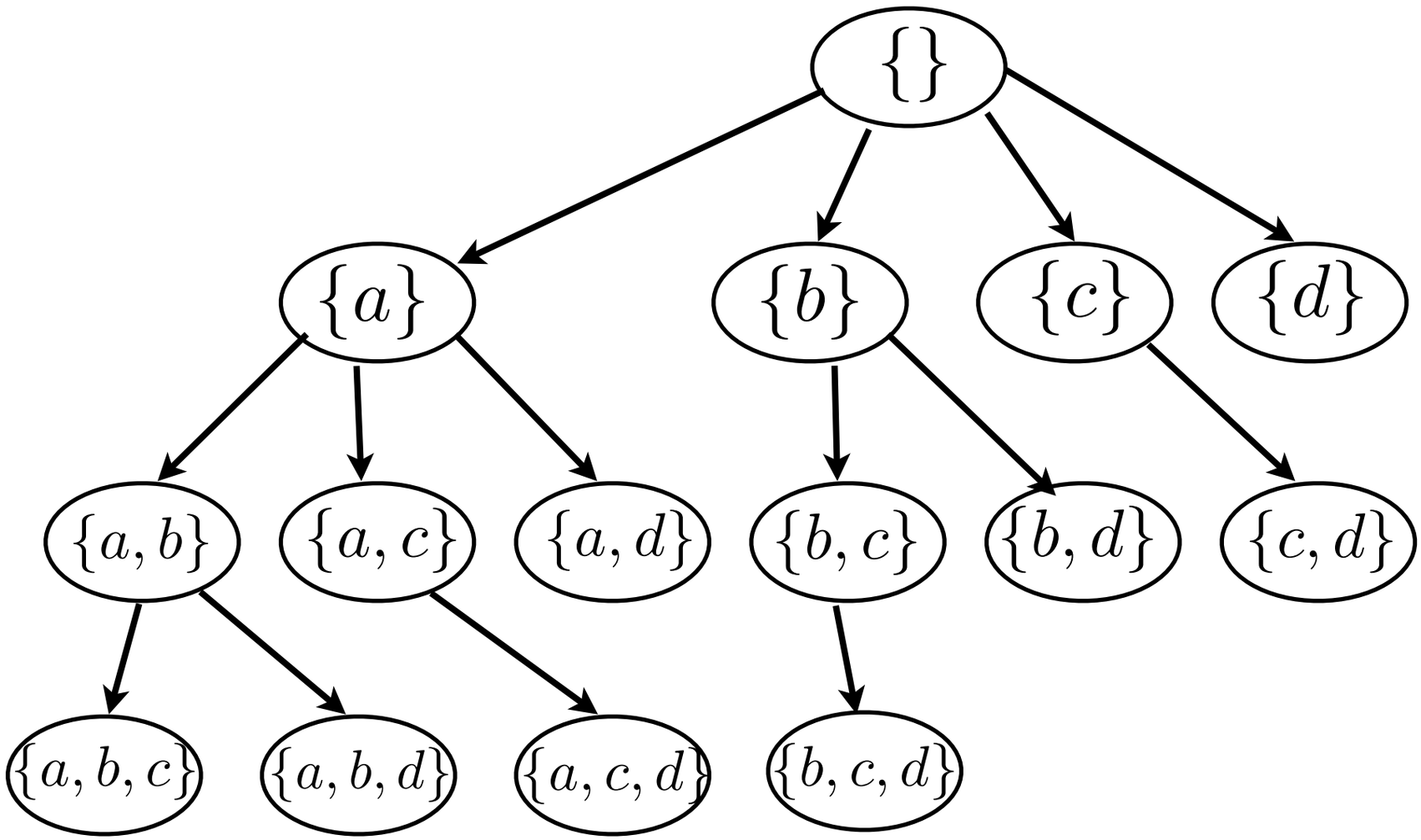
step 1

step 2

# Algorithm PaSAL

property with anti-monotonicity can be used for pruning

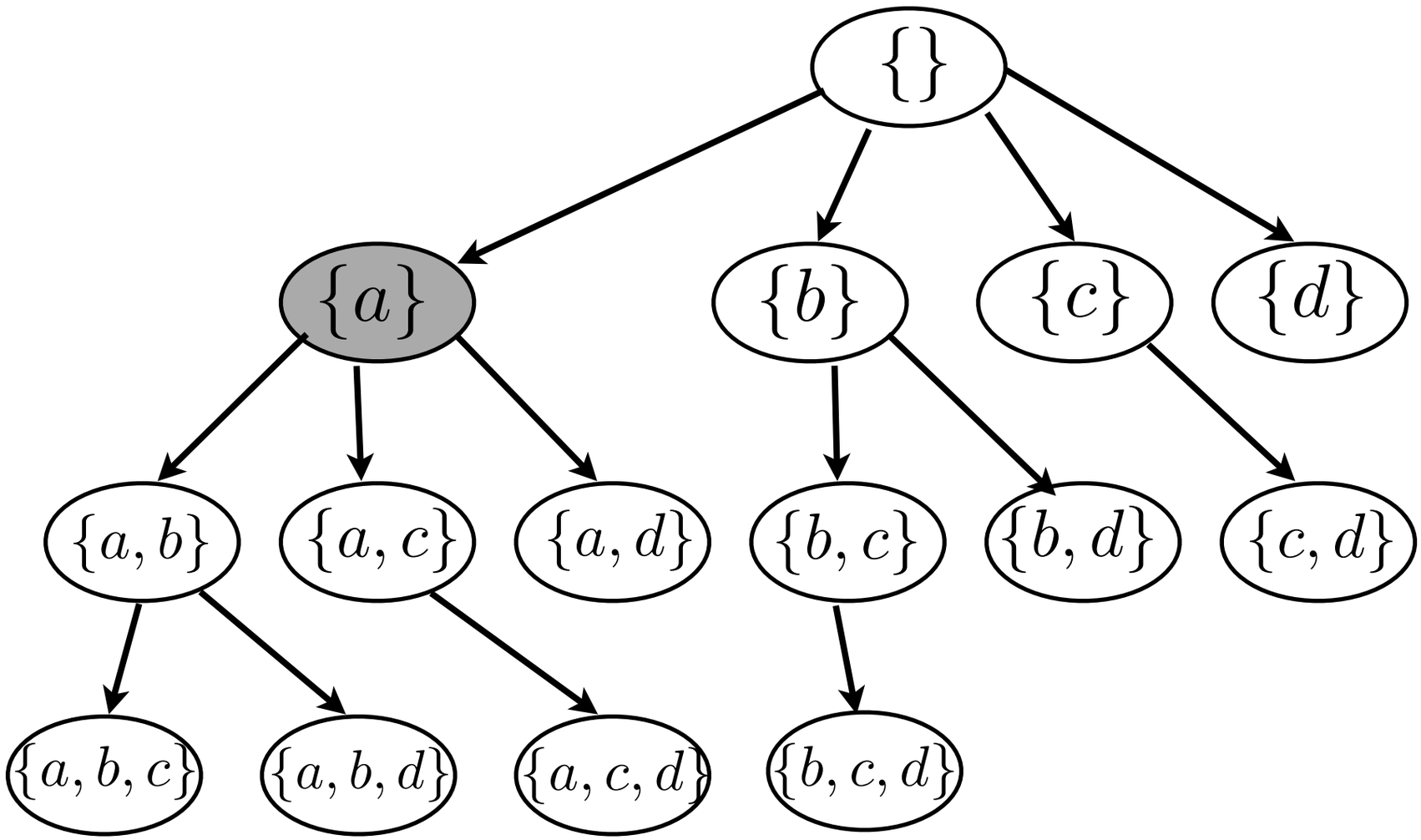
CM meets the requirement



# Algorithm PaSAL

property with anti-monotonicity can be used for pruning

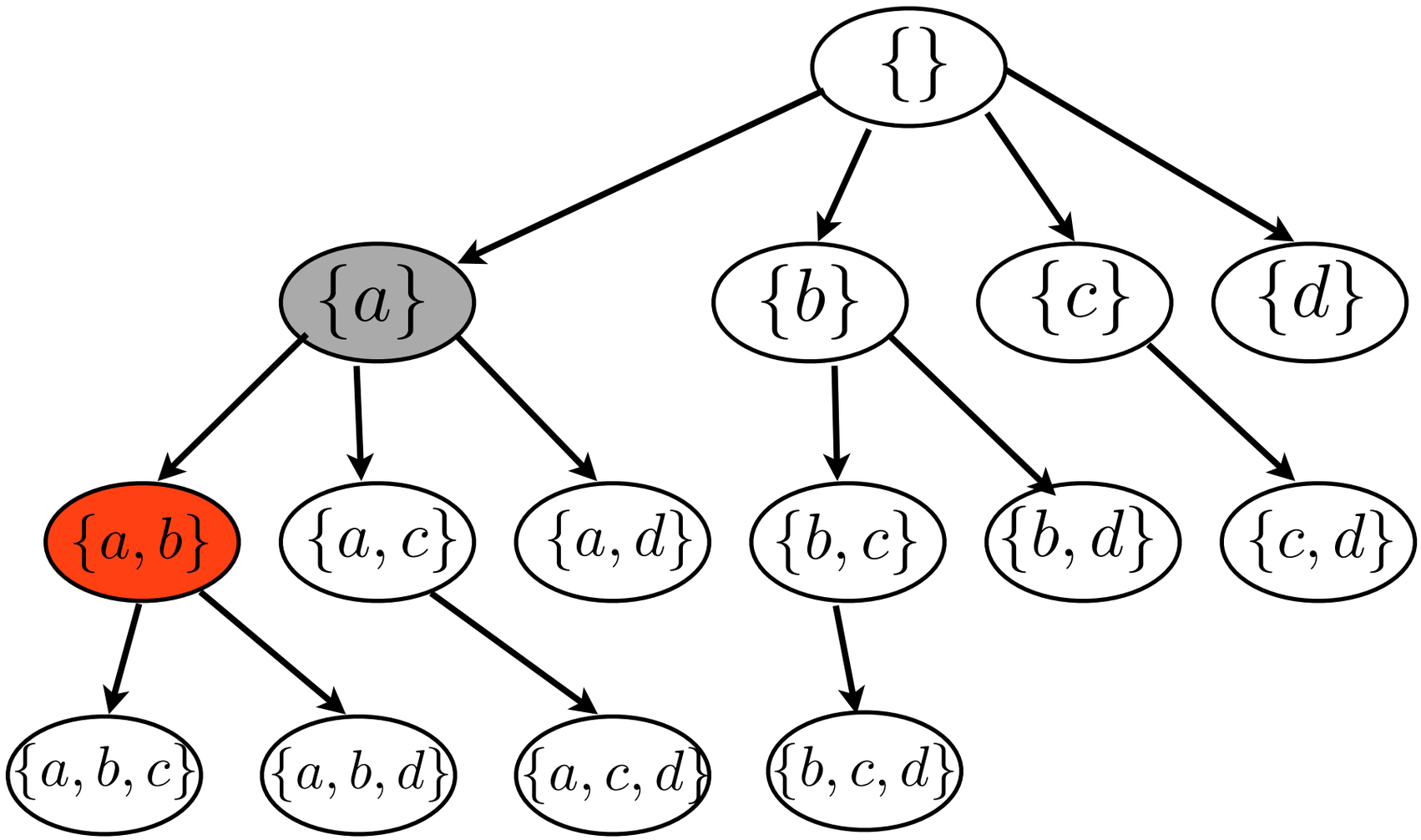
CM meets the requirement



# Algorithm PaSAL

property with anti-monotonicity can be used for pruning

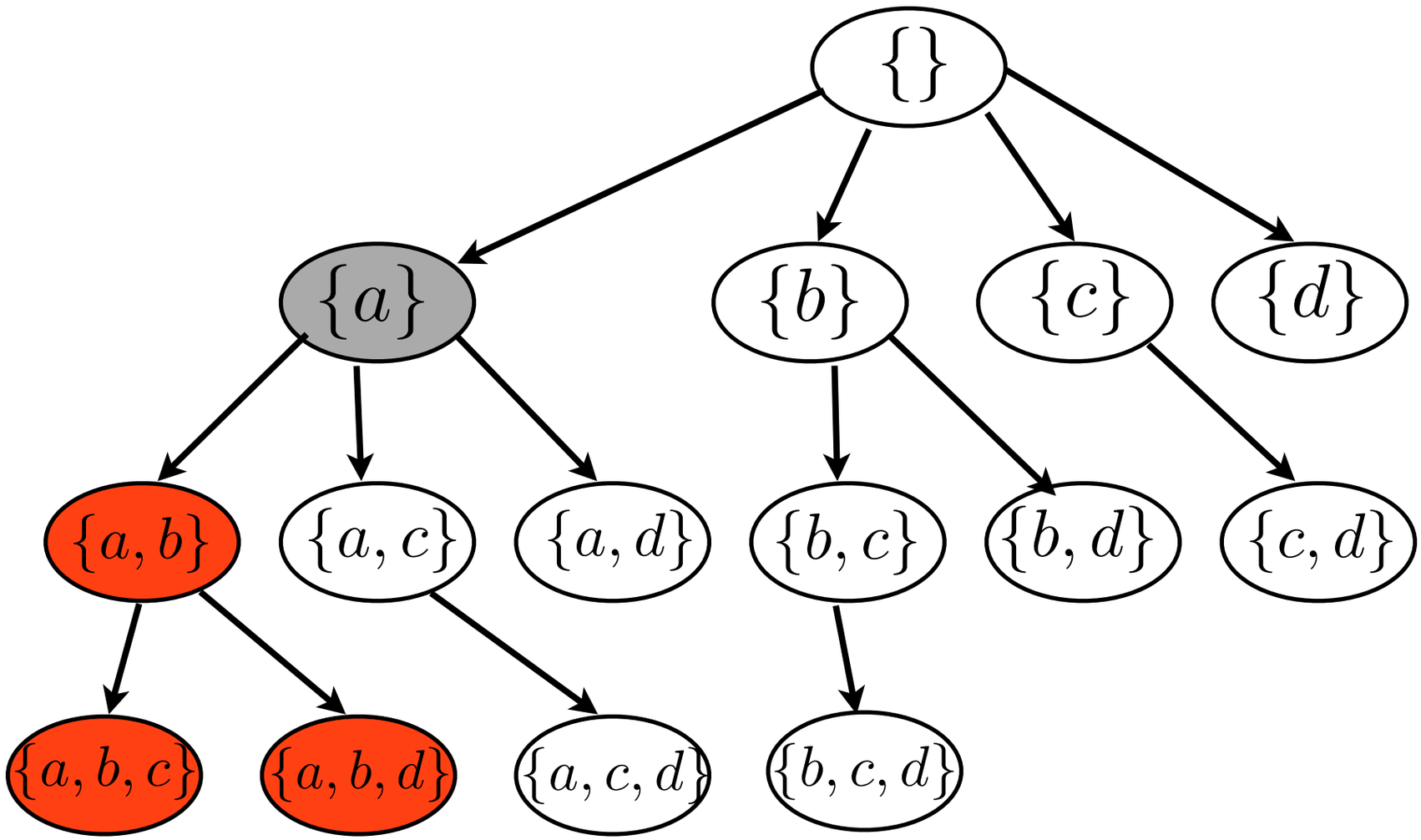
CM meets the requirement



# Algorithm PaSAL

property with anti-monotonicity can be used for pruning

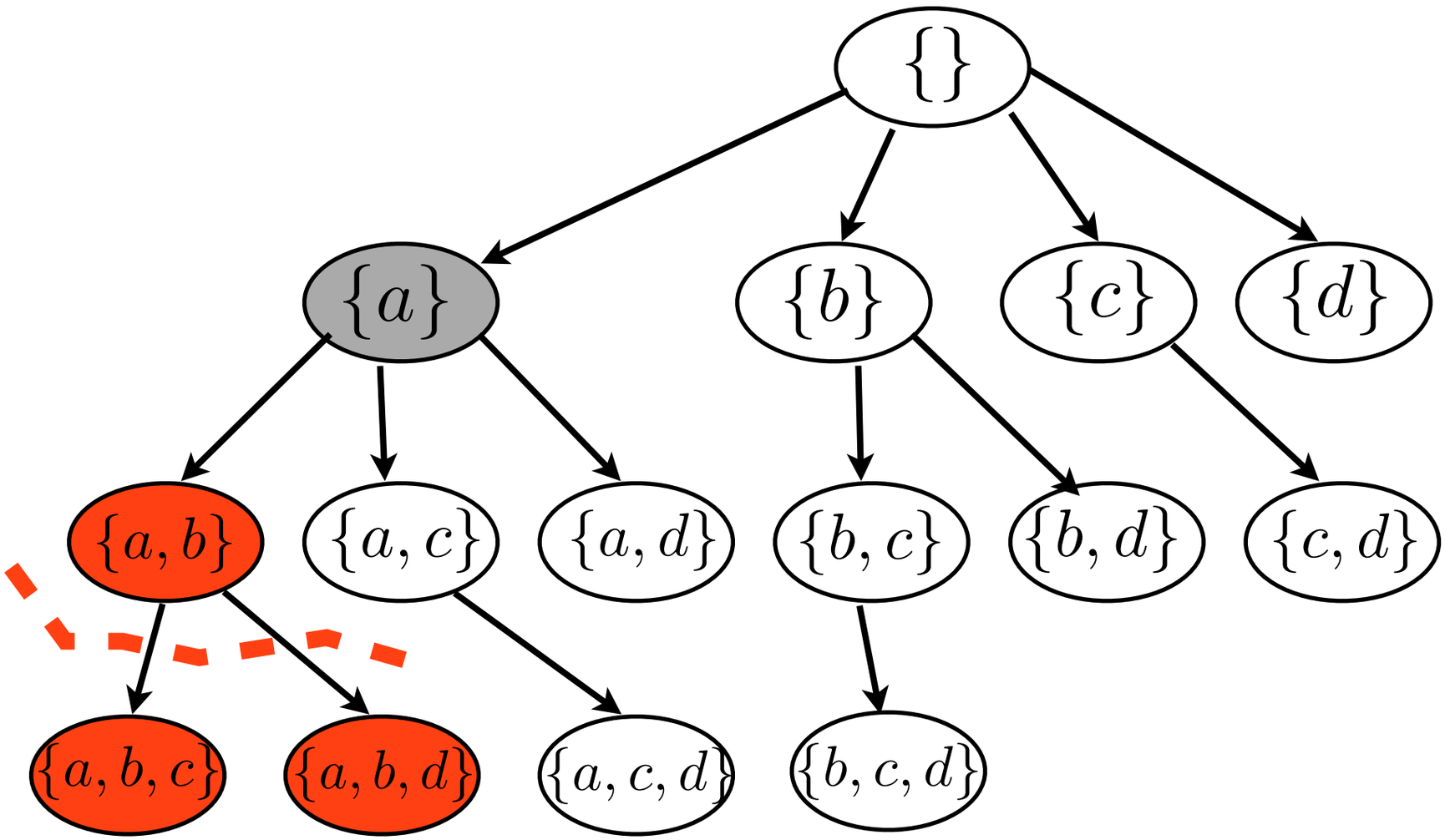
CM meets the requirement



# Algorithm PaSAL

property with anti-monotonicity can be used for pruning

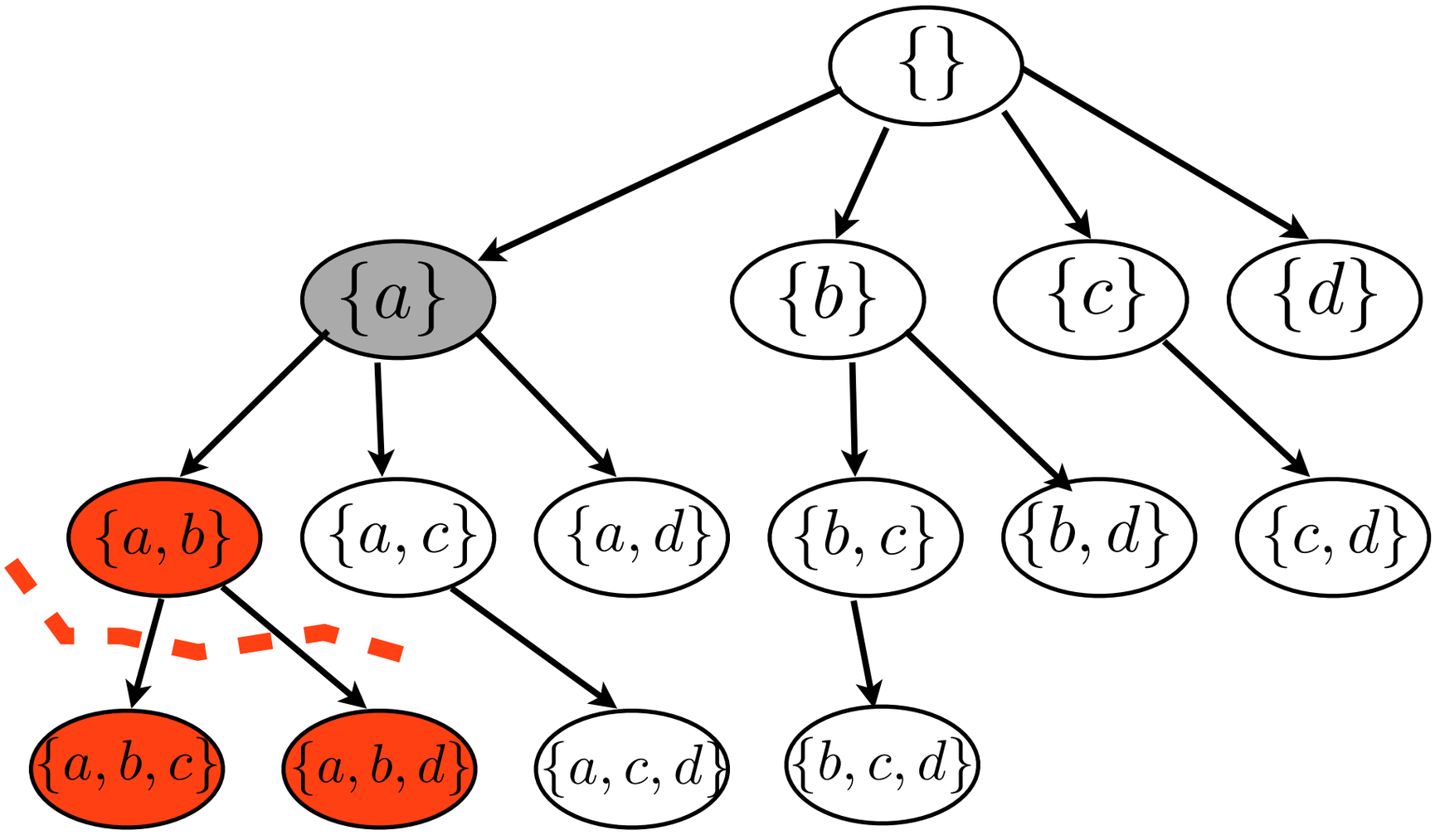
CM meets the requirement



# Algorithm PaSAL

property with anti-monotonicity can be used for pruning

CM meets the requirement




**in-process:**

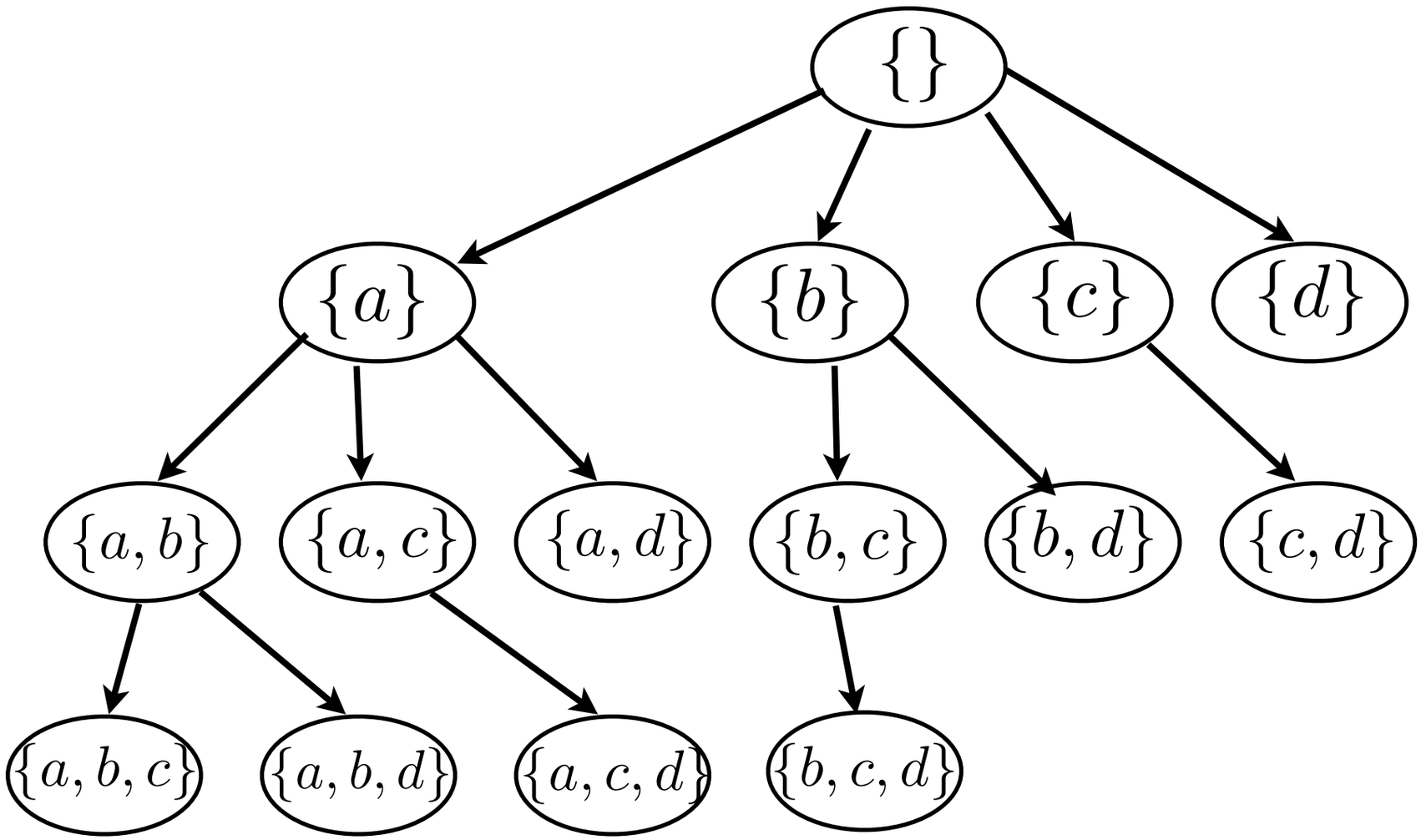
introduce CM to reduce the search space

# Algorithm PaSAL

$a = \{1, 1, 0\}$

$b = \{0, 1, 0\}$

$c = \{1, 0, 0\}$



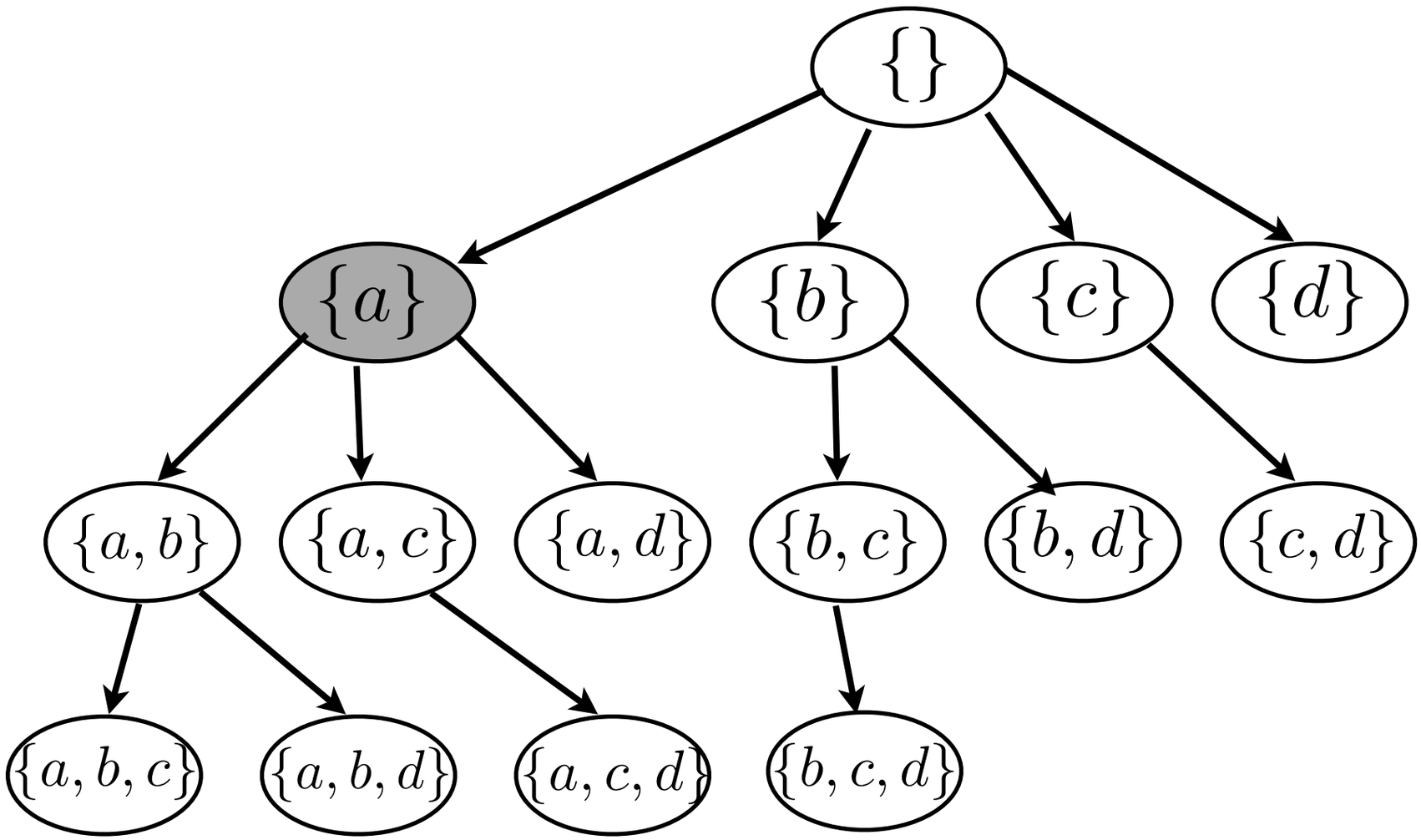
# Algorithm PaSAL

$a = \{1, 1, 0\}$

$b = \{0, 1, 0\}$

$c = \{1, 0, 0\}$

$\{1, 1, 0\}$

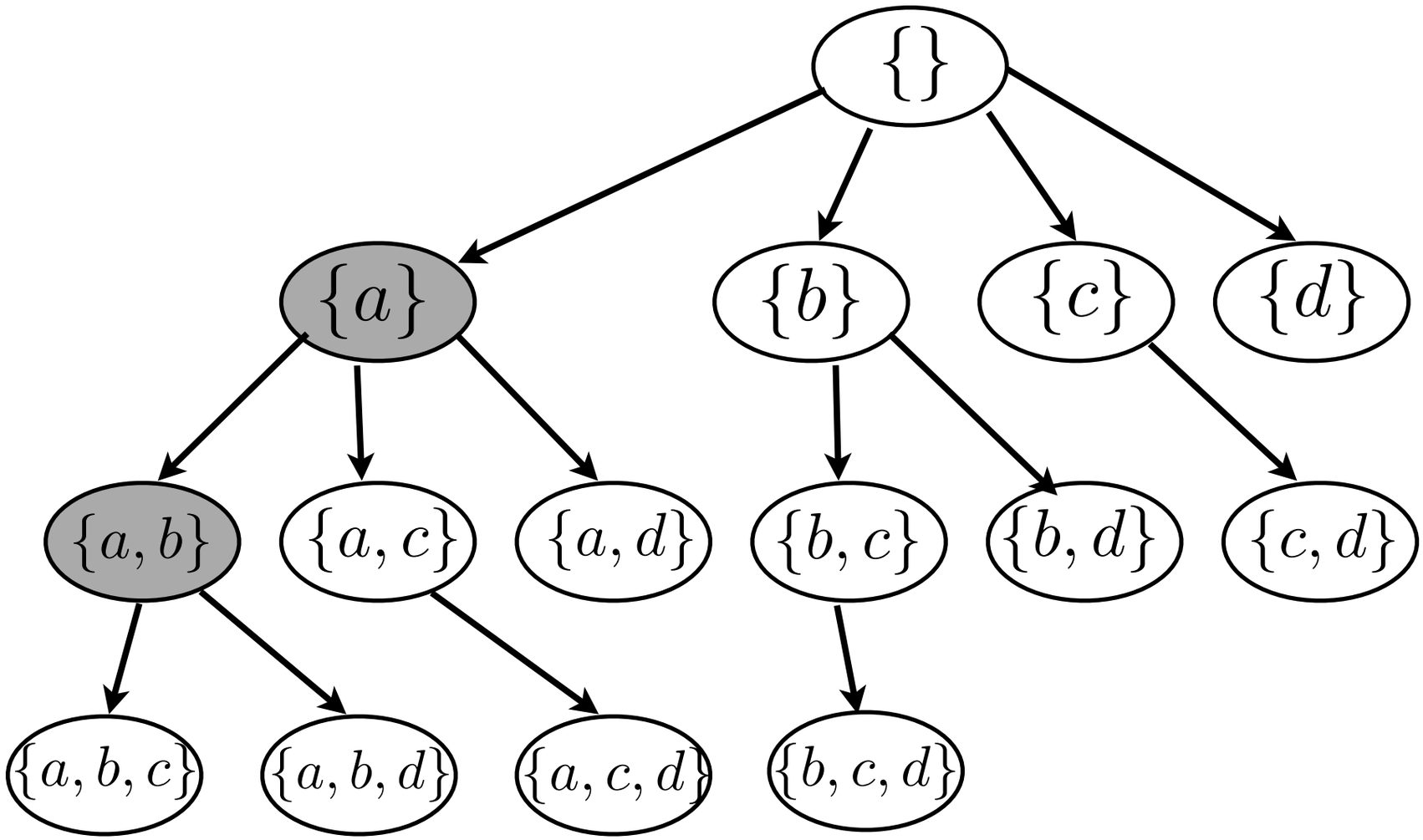


# Algorithm PaSAL

$a = \{1, 1, 0\}$ $\{1, 1, 0\}$

$b = \{0, 1, 0\}$ $\{0, 1, 0\}$

$c = \{1, 0, 0\}$



# Algorithm PaSAL

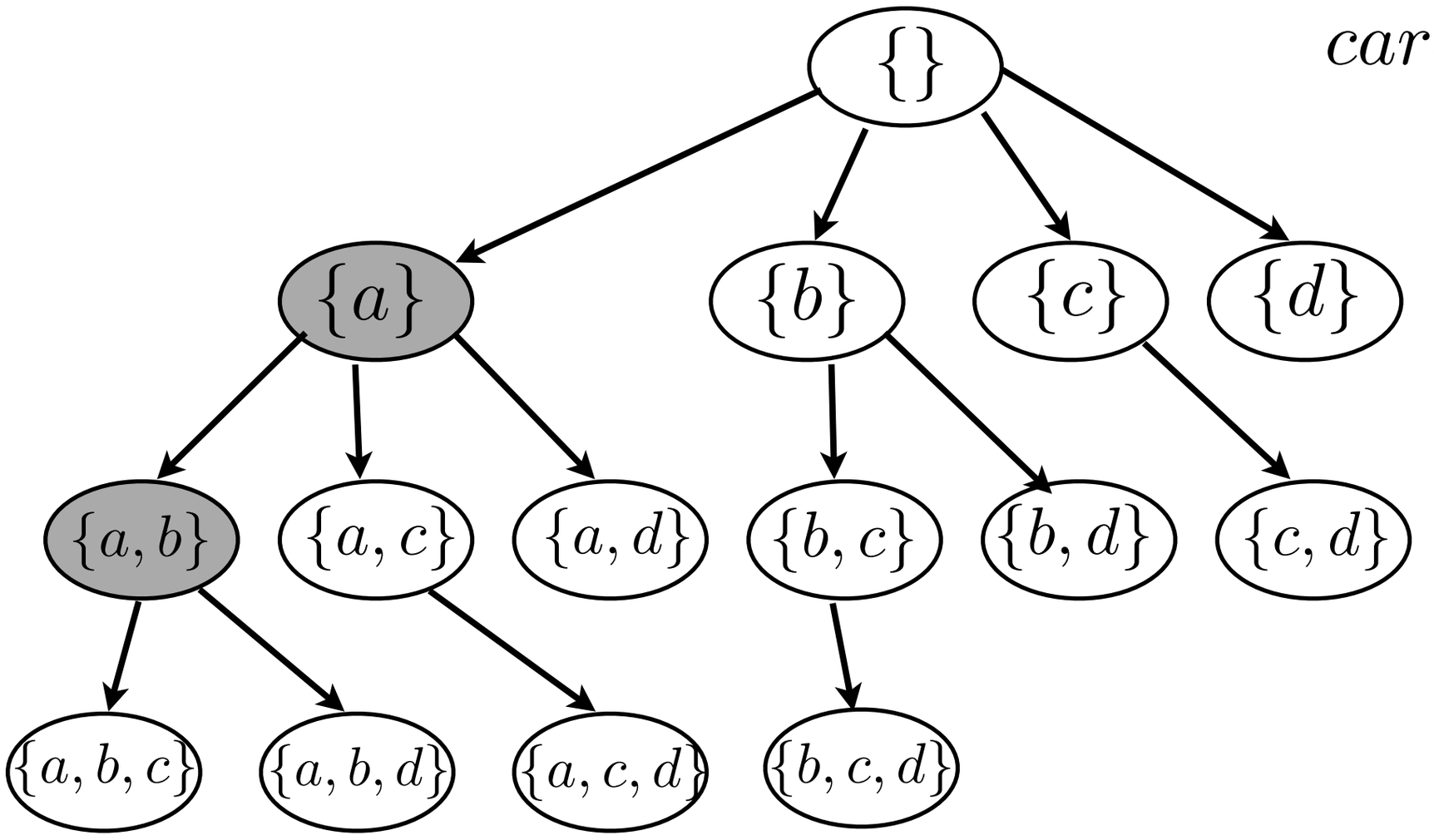
$a = \{1, 1, 0\}$

$b = \{0, 1, 0\}$

$c = \{1, 0, 0\}$

$$\begin{array}{rr} & \{1,1,0\} \\ \text{AND} & \{0,1,0\} \\ \hline & \{0,1,0\} \end{array}$$

$card(\{0, 1, 0\}) = 1 \neq 0$



# Algorithm PaSAL

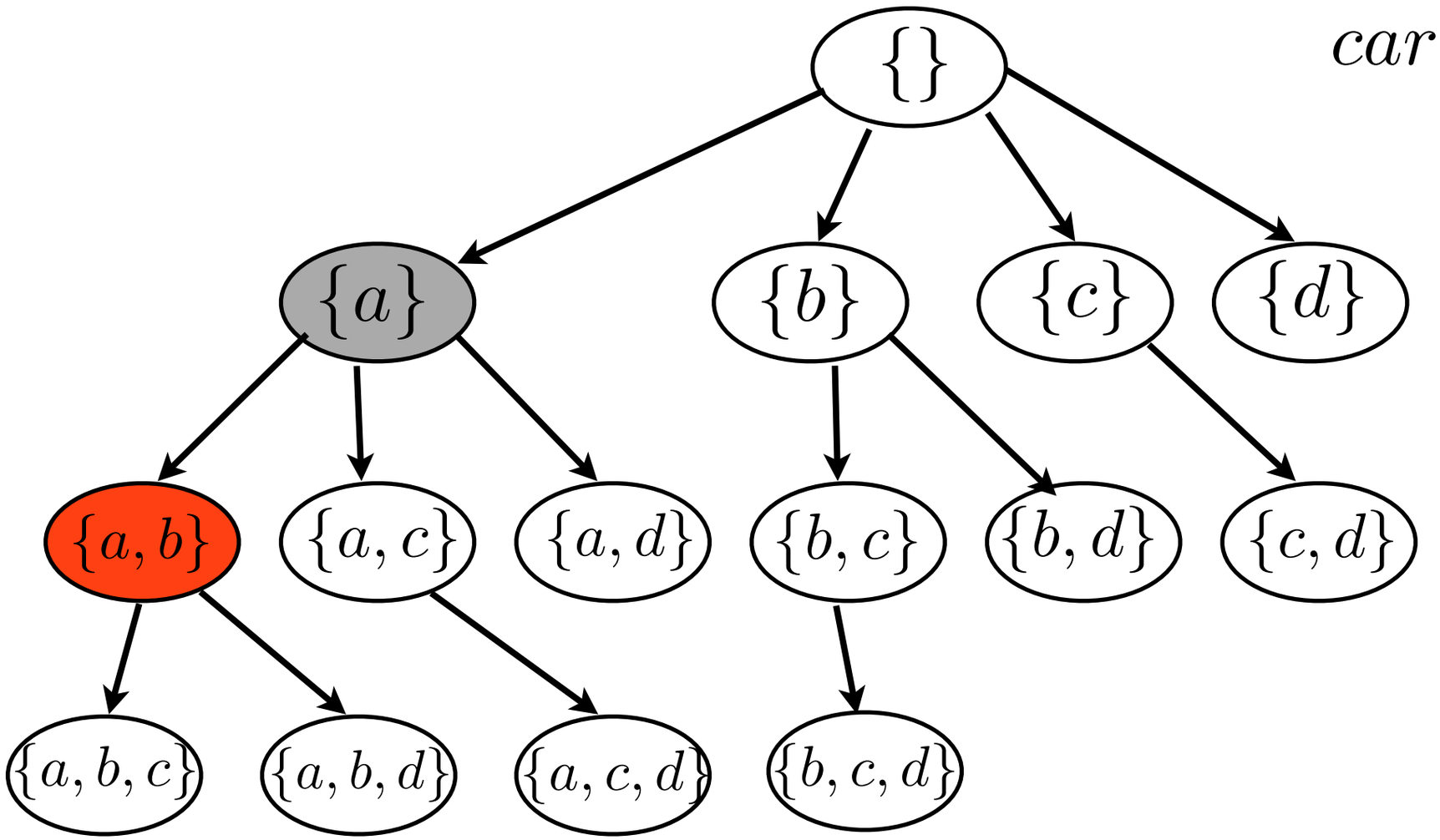
$a = \{1, 1, 0\}$

$b = \{0, 1, 0\}$

$c = \{1, 0, 0\}$

$$\begin{array}{rr} & \{1,1,0\} \\ \text{AND} & \{0,1,0\} \\ \hline & \{0,1,0\} \end{array}$$

$card(\{0, 1, 0\}) = 1 \neq 0$



# Algorithm PaSAL

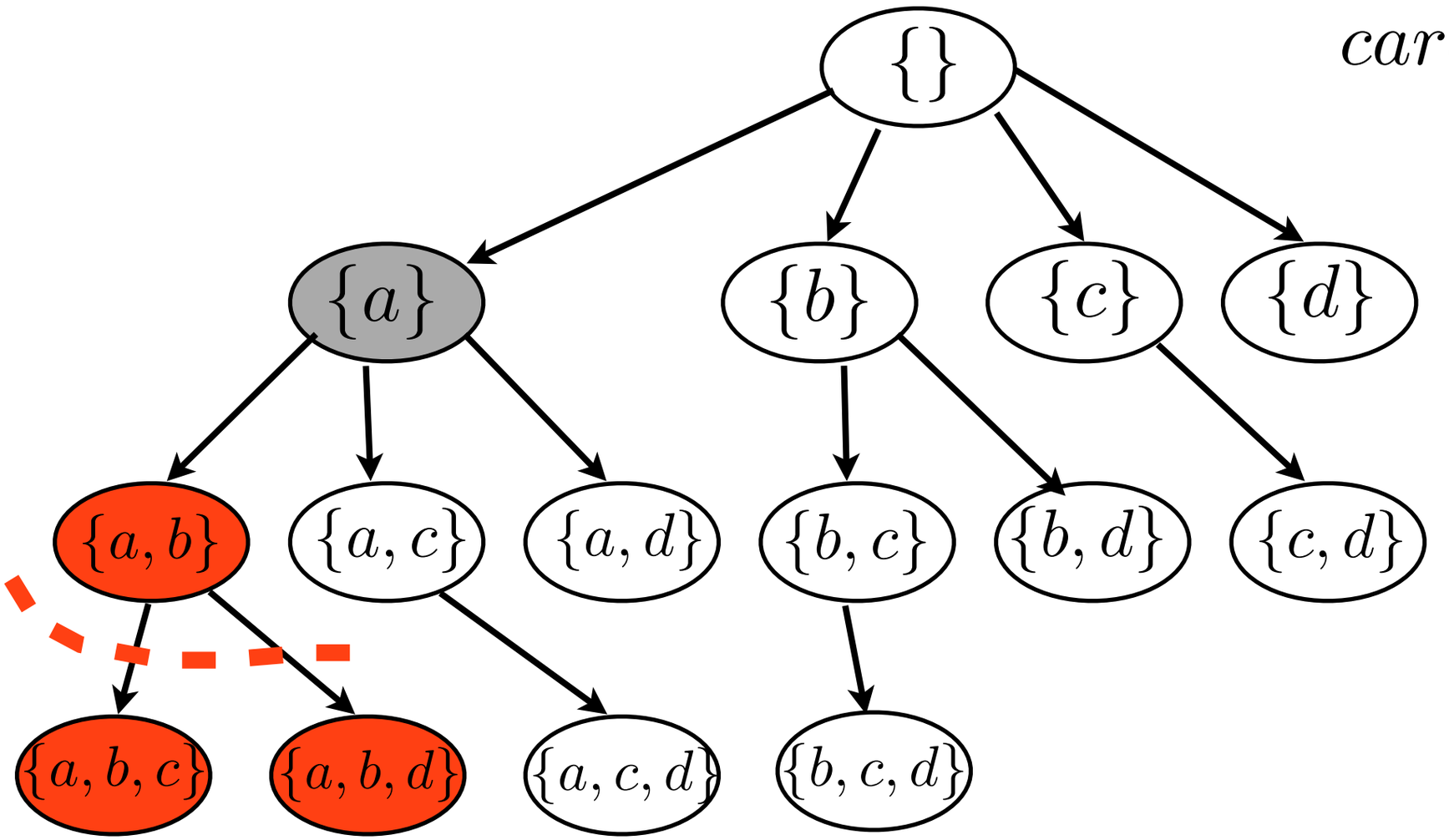
$a = \{1, 1, 0\}$

$b = \{0, 1, 0\}$

$c = \{1, 0, 0\}$

$$\begin{array}{rr} & \{1,1,0\} \\ \text{AND} & \{0,1,0\} \\ \hline & \{0,1,0\} \end{array}$$

$card(\{0, 1, 0\}) = 1 \neq 0$



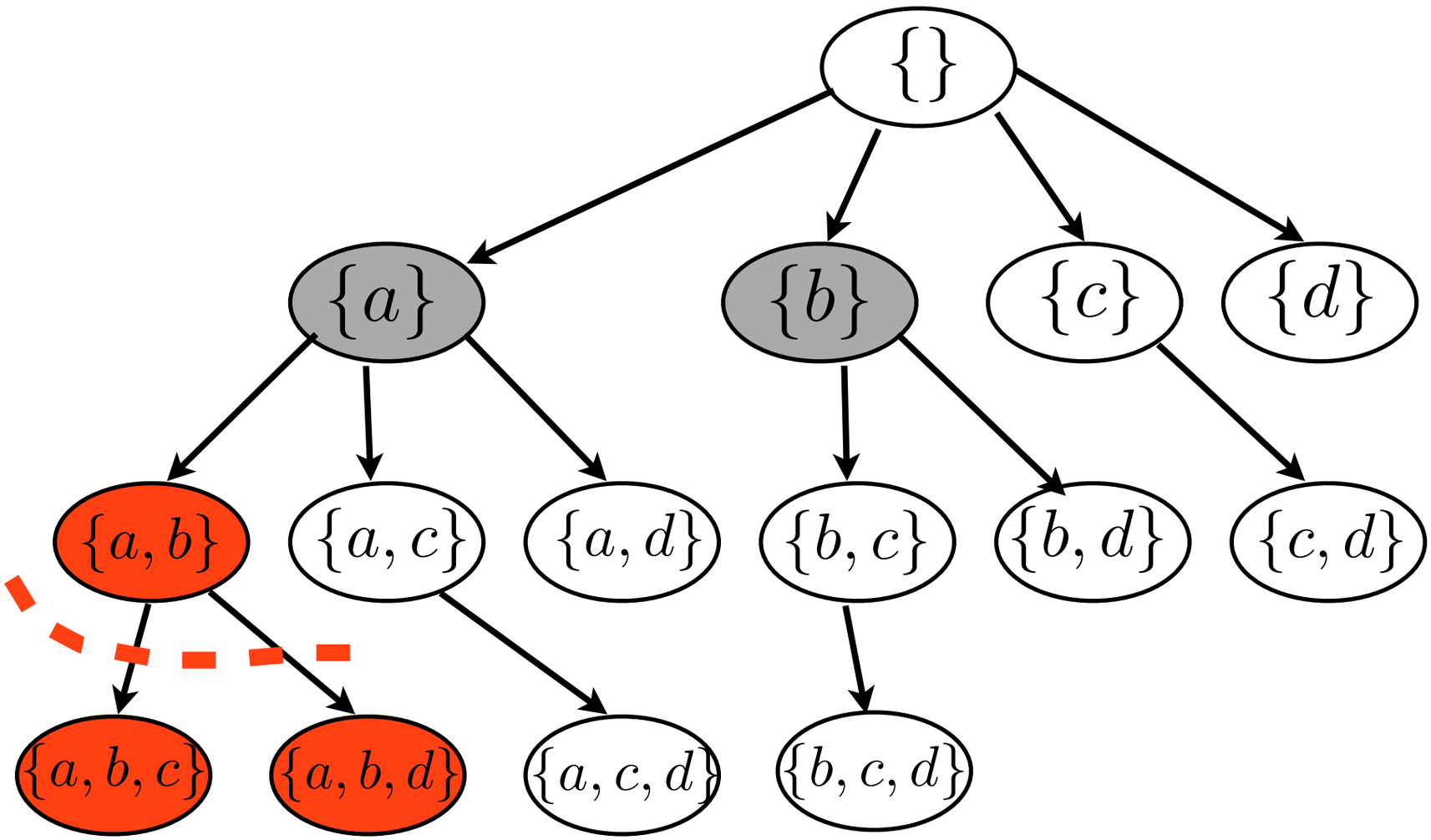
# Algorithm PaSAL

$a = \{1, 1, 0\}$

$b = \{0, 1, 0\}$

$c = \{1, 0, 0\}$

$\{0, 1, 0\}$

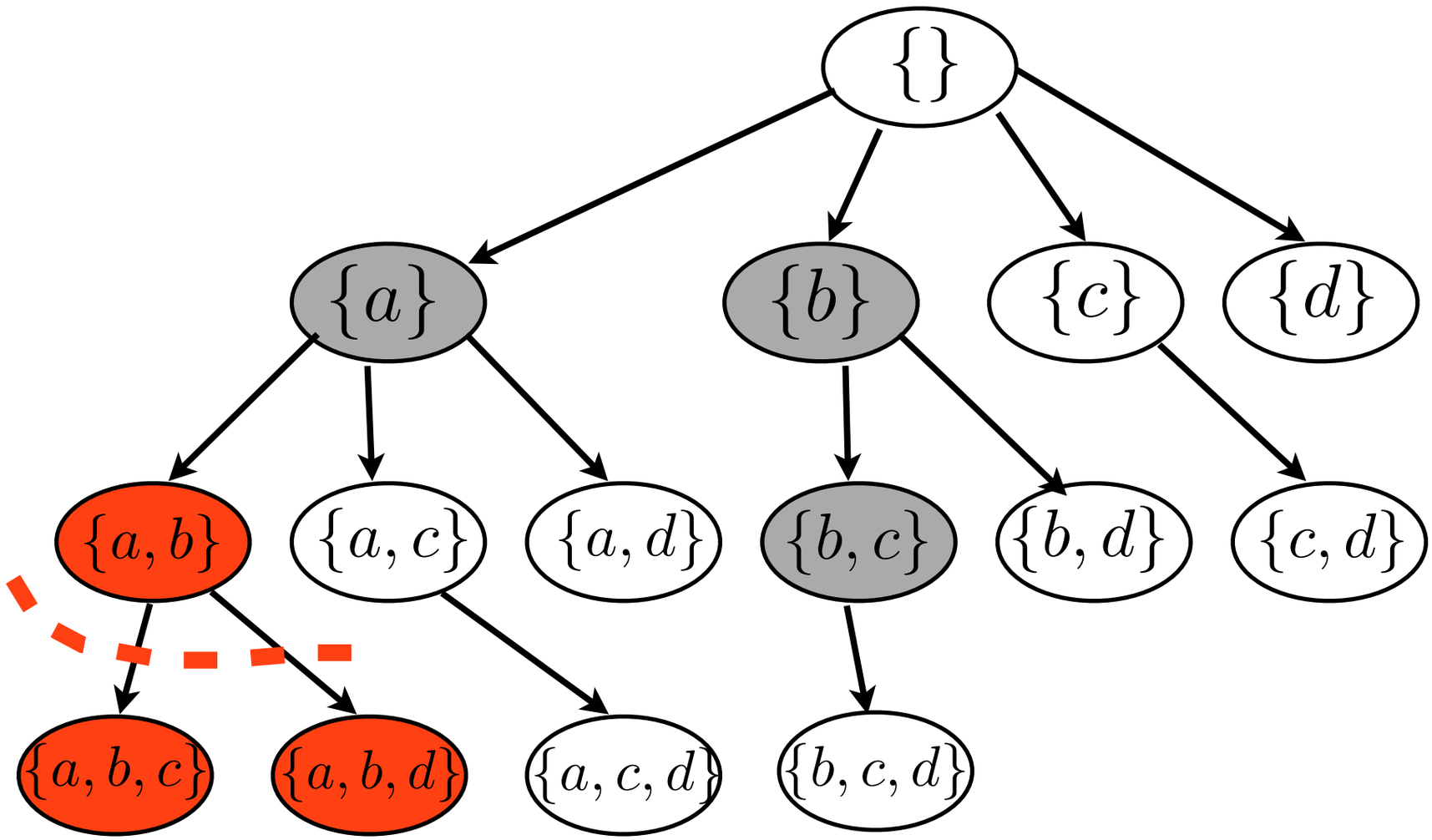


# Algorithm PaSAL

$a = \{1, 1, 0\}$ $\{0, 1, 0\}$

$b = \{0, 1, 0\}$ $\{1, 0, 0\}$

$c = \{1, 0, 0\}$



# Algorithm PaSAL

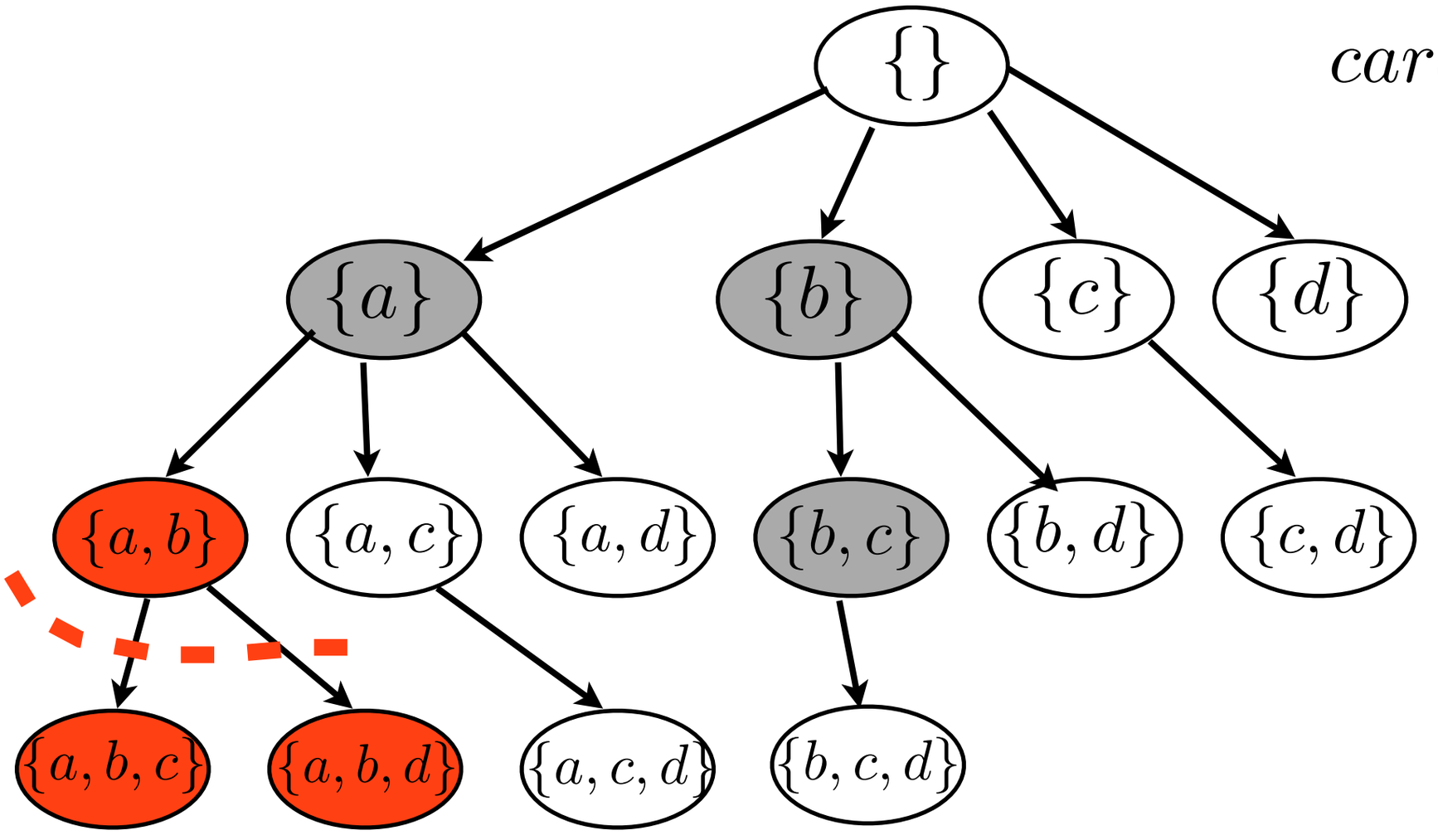
$a = \{1, 1, 0\}$

$b = \{0, 1, 0\}$

$c = \{1, 0, 0\}$

$$\begin{array}{rr} & \{0,1,0\} \\ \text{AND} & \{1,0,0\} \\ \hline & \{0,0,0\} \end{array}$$

$card(\{0, 0, 0\}) = 0$



# Algorithm PaSAL

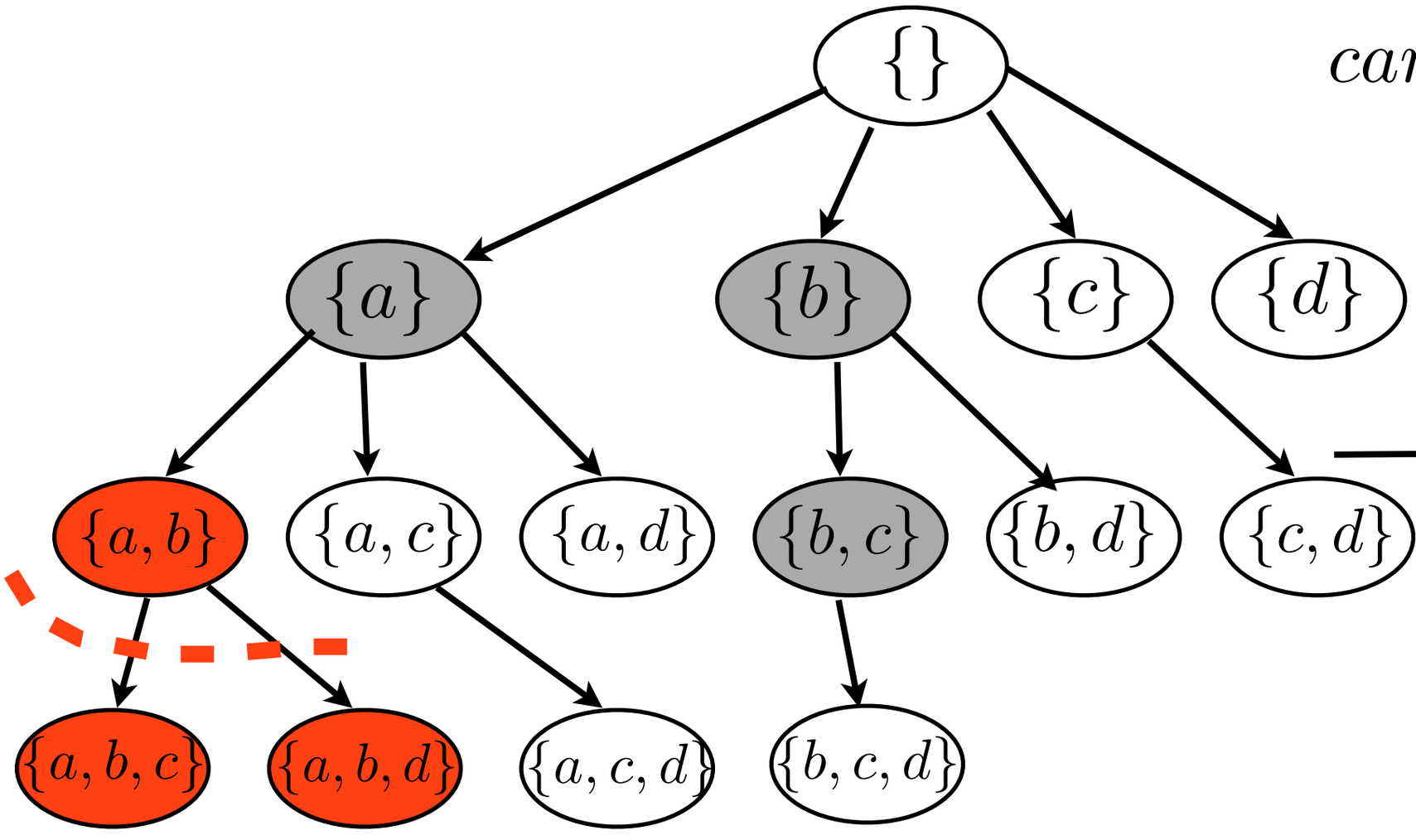
$a = \{1, 1, 0\}$

$b = \{0, 1, 0\}$

$c = \{1, 0, 0\}$

$$\begin{array}{rr} & \{0,1,0\} \\ \text{AND} & \{1,0,0\} \\ \hline & \{0,0,0\} \end{array}$$

$card(\{0, 0, 0\}) = 0$




$$\begin{array}{rr} & \{0,1,0\} \\ \text{OR} & \{1,0,0\} \\ \hline & \{1,1,0\} \end{array}$$

# Algorithm PaSAL

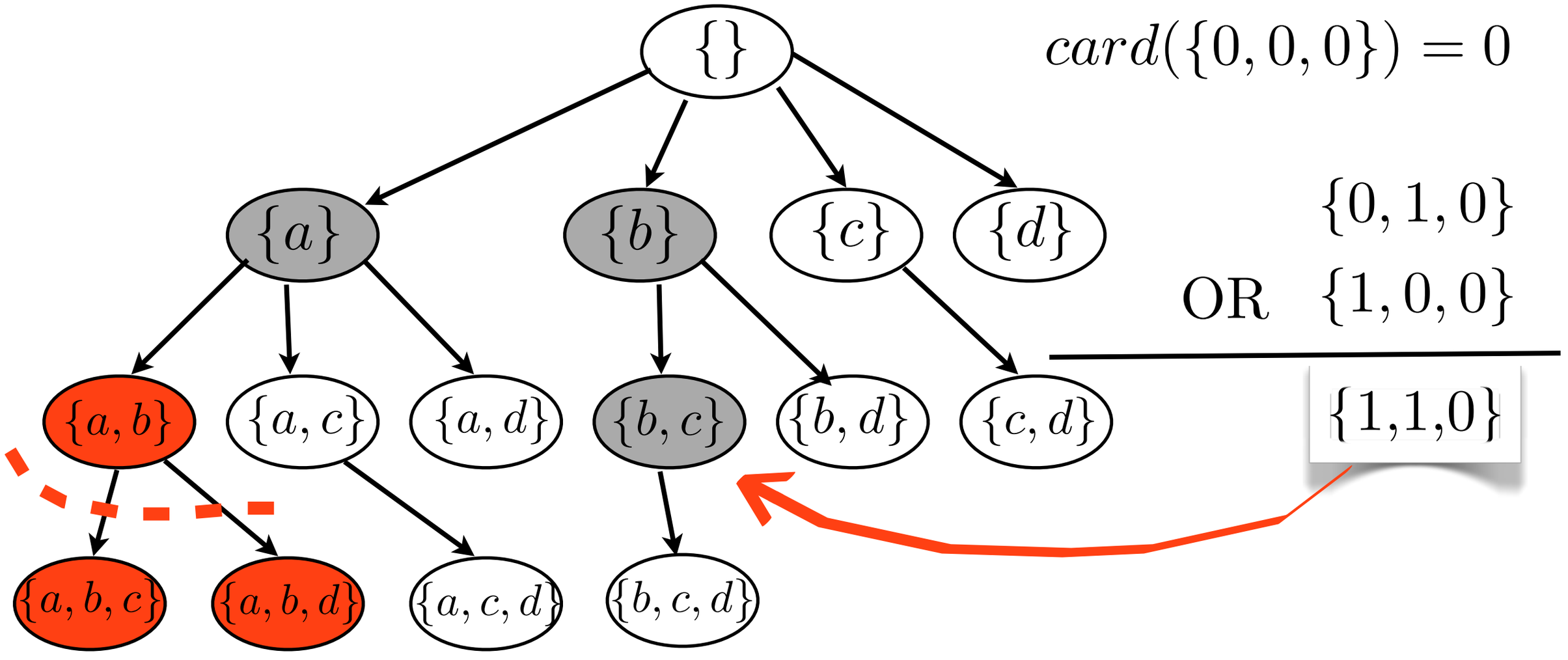
$a = \{1, 1, 0\}$

$b = \{0, 1, 0\}$

$c = \{1, 0, 0\}$

$$\begin{array}{rr} & \{0,1,0\} \\ \text{AND} & \{1,0,0\} \\ \hline & \{0,0,0\} \end{array}$$

$card(\{0, 0, 0\}) = 0$




$$\begin{array}{rr} & \{0,1,0\} \\ \text{OR} & \{1,0,0\} \\ \hline & \{1,1,0\} \end{array}$$

# Algorithm PaSAL

**Algorithm 2:** PaSAL

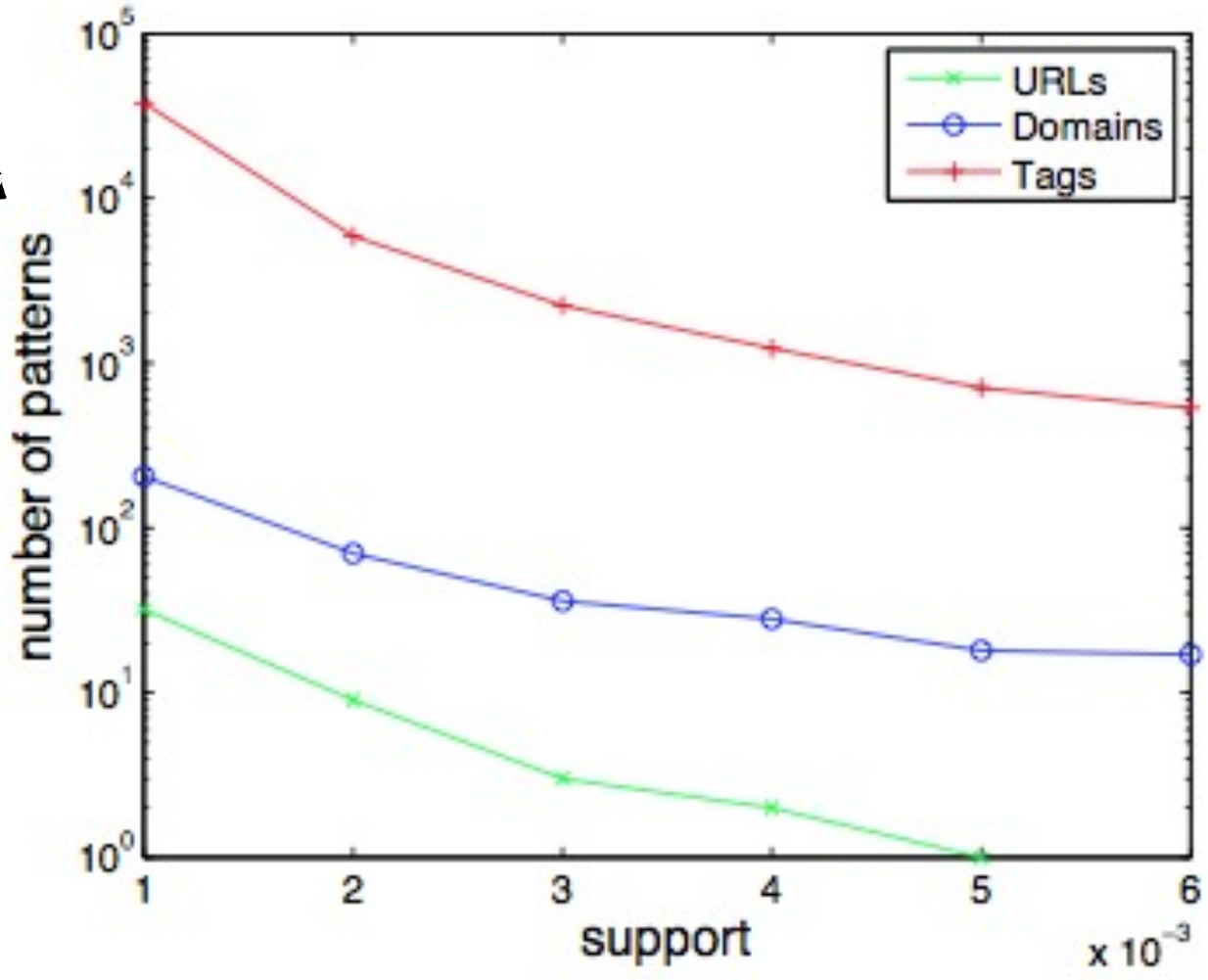
**input** : the tag transaction data $\mathcal{T}^{trans}$, lexicographic ordering $\mathcal{T}$, the conflict matrix $CM$
**output**: restrict condition constrained patterns $\mathcal{P}$

1 $Root.head \leftarrow empty$;
2 $Root.tail \leftarrow \mathcal{T}$;
3 $Root.flag \leftarrow (0)_2$;
4 $DFS(Root)$;
5 $DFS(Node)$ **begin**
6  **for** $tag \in Node.tail$ **do**
7  $p_{tmp} \leftarrow Node.head \cup tag$;
8  $flag_{tmp} \leftarrow (Node.flag \,\&\, CM_{tag})_2$;
9  **if** $p_{tmp}.frequency > min\_sup \times |\mathcal{T}^{trans}|$ ***and*** $flag_{tmp} == 0$ **then**
10  $Node_{tmp}.head \leftarrow p_{tmp}$;
11  remove tag from $Node_{tmp}.tail \leftarrow Node.tail$;
12  $Node_{tmp}.flag \leftarrow (Node.flag \mid CM_{tag})_2$;
13  $Children \leftarrow Node_{tmp}$;
14  $P_{candidate} \leftarrow Node_{tmp}.head$;
15  **for** $node \in Children$ **do**
16  $DFS(node)$;
17 $P \leftarrow P_{candidate}$;

# Experiment: effectiveness

| Database Name | Number of Records | Number of Transactions |
|---|---|---|
| URLs | 107031 | 23212 |
| Domains | 59416 | 23211 |
| Tags | 328752 | 16041 |

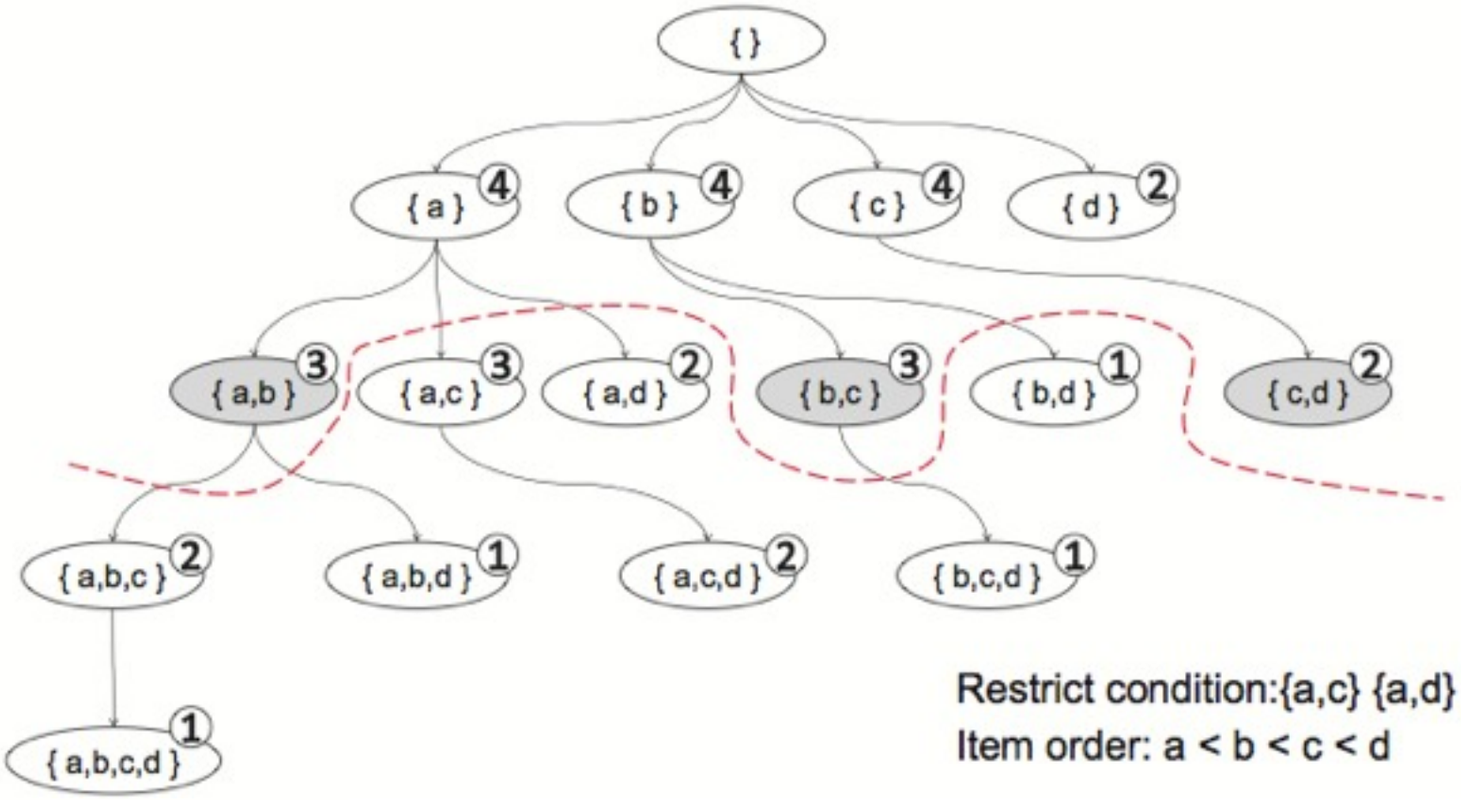
logarithm

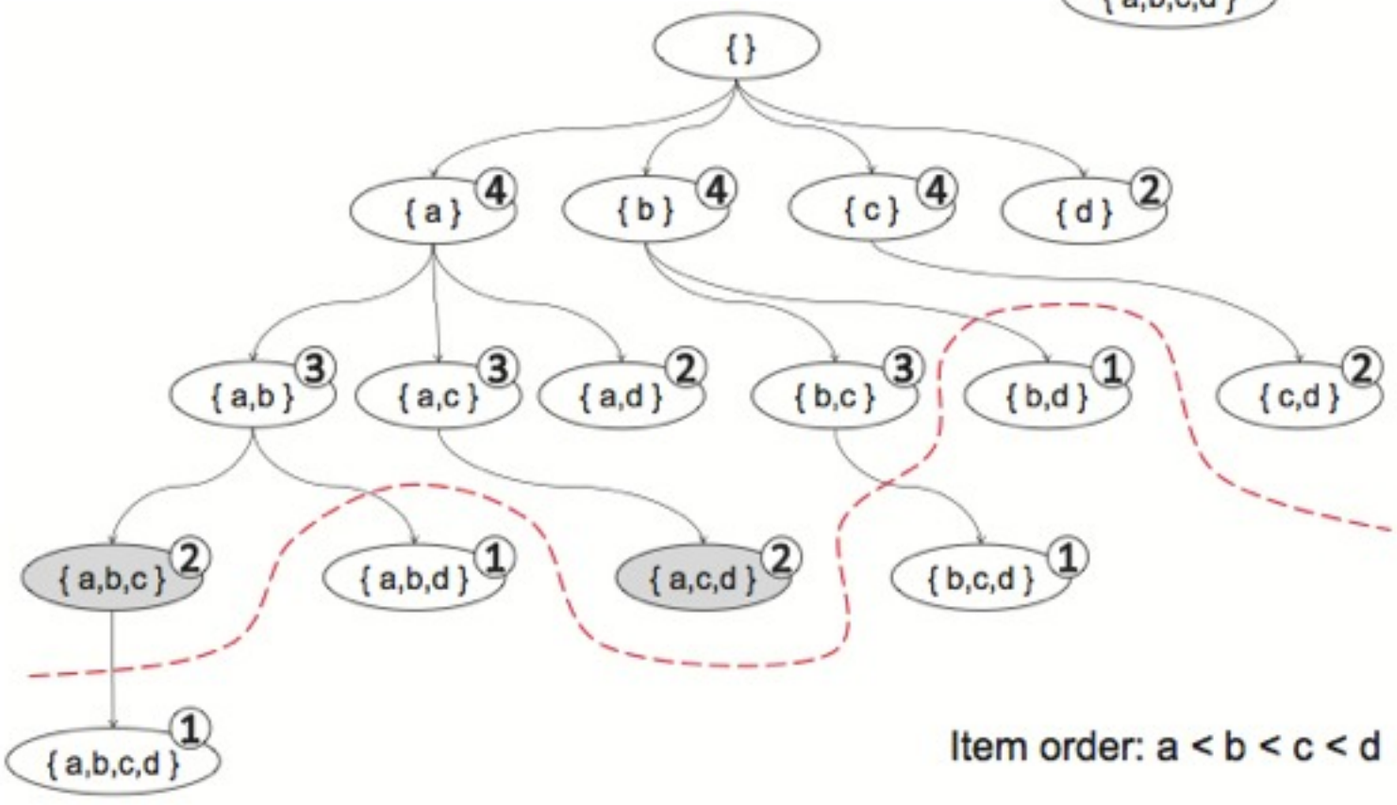



By using tags, we enrich the patterns

# Experiment: efficiency

| No. | Items |
|---|---|
| $t_1$ | b c |
| $t_2$ | a b |
| $t_3$ | a b c |
| $t_4$ | a b d |
| $t_5$ | a b c d |



# Experiment: efficiency

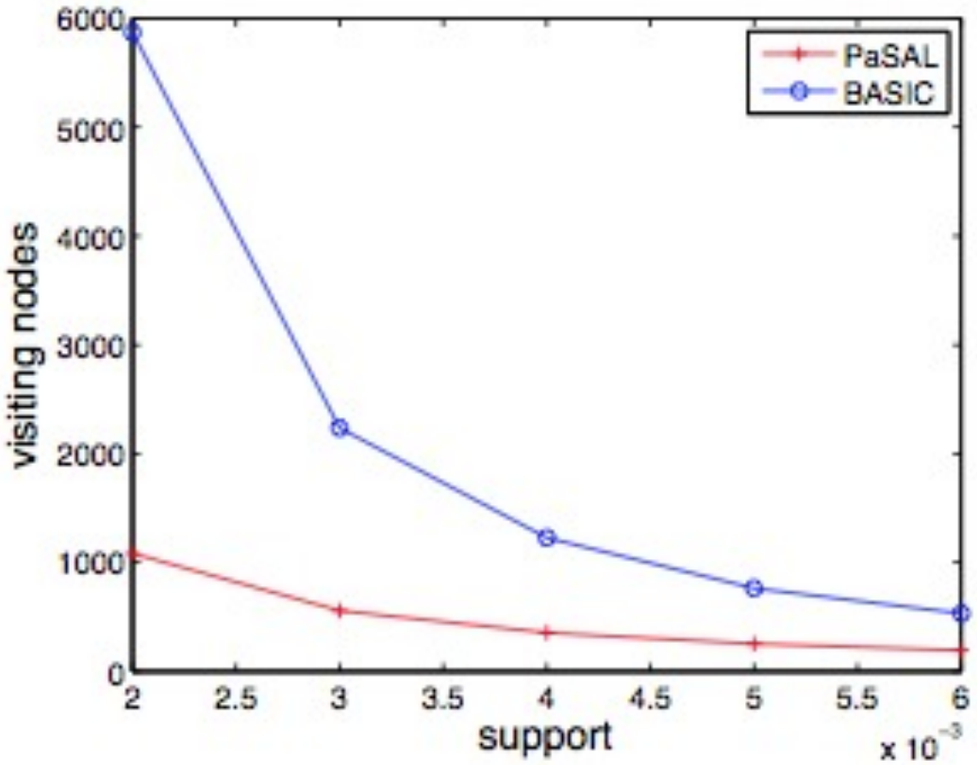



**Fig. 4.** Visiting Nodes

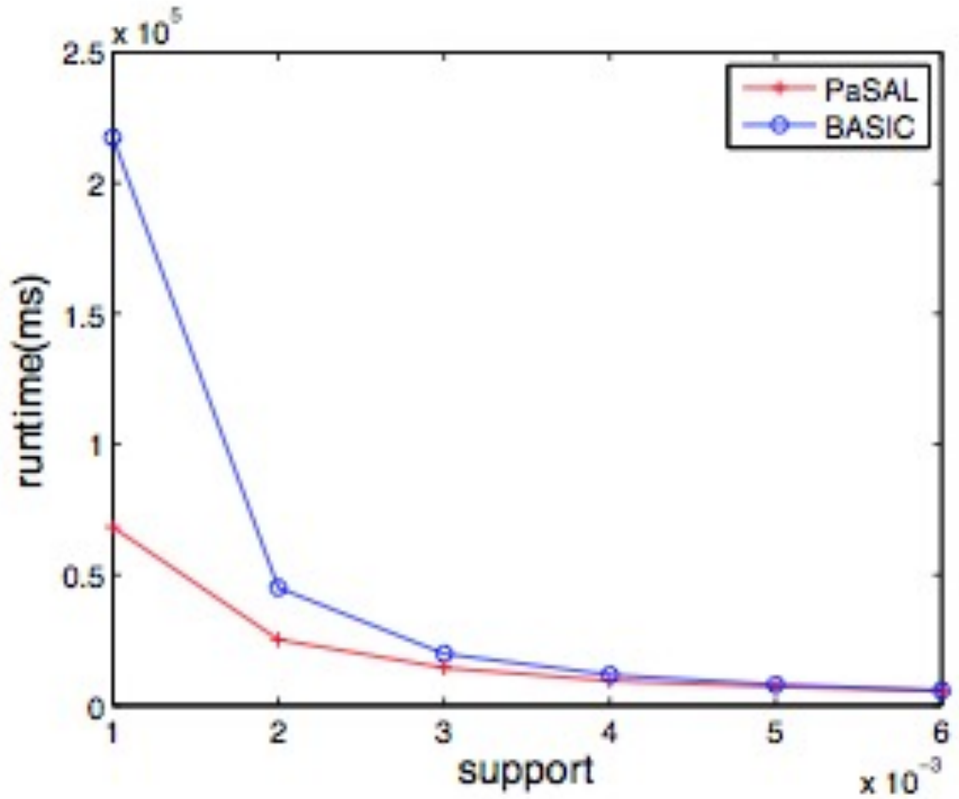



**Fig. 5.** Running Time

PaSAL outperforms the Basic algorithm

# Conclusion

define a novel frequent pattern mining problem by exploring print logs

expand semantics of each url with external thesaurus by leveraging delicious.com api

design an efficient algorithm PaSAL to mine frequent patterns with semantically alternative labels