Organizational matters

- Final exam: Tuesday, 19 February, at twelve o’clock noon
  - Same room (might change later)
- Re-exam: Tuesday, 19 March, at twelve o’clock noon
- Guideline on returning the essay now on-line
  - Your name, matriculation number & e-mail address **must** be in every essay
  - Also essay topic must be clearly written
  - **Only PDFs**
  - Please start the e-mail subject with ”DTDM” and have word ”essay” somewhere in it

More organization

- **Registration to the final exam in HISPOS**
  - DL: 4th of November
  - Can cancel until two weeks before final exam
  - Contact study office in case of problems
- The lecture on 27th of November might get cancelled
  - Will postpone the schedule by one week
  - Will be confirmed next week w/ more info about changes in the schedule

Topic I: Pattern Set Mining

Discrete Topics in Data Mining
Universität des Saarlandes, Saarbrücken
Winter Semester 2012/13

Introduction to Pattern (Set) Mining

1. What is Pattern Mining
2. Frequent Itemsets
    2.1. Downwards closedness property
    2.2. The Apriori Algorithm
3. The Flood of Itemsets
    3.1. Closed, Maximal & Non-Derivable Itemsets
4. Global and Local Data Mining

Z & M, Ch. 8 & 9; T, S & K, Ch. 6

Pattern Mining

- Pattern mining is about finding *patterns* from the data
- But what are the patterns?
  - Frequent itemsets (method-oriented)
  - Any repeated (or anomalous) activity in the data
- US National Research Council says
  - *Pattern-based data mining looks for patterns (including anomalous data patterns) that might be associated with terrorist activity — these patterns might be regarded as small signals in a large ocean of noise.*

Frequent Itemsets

- **Frequent itemsets** are an important concept in pattern mining
  - Many other concepts are defined based on them
  - We'll meet these concepts a bit later
  - Yet, we'll see they're not without their faults…
- Mining all frequent itemsets was all the rage back in Nineties and early millenium
- An itemset is defined over **transactional** database

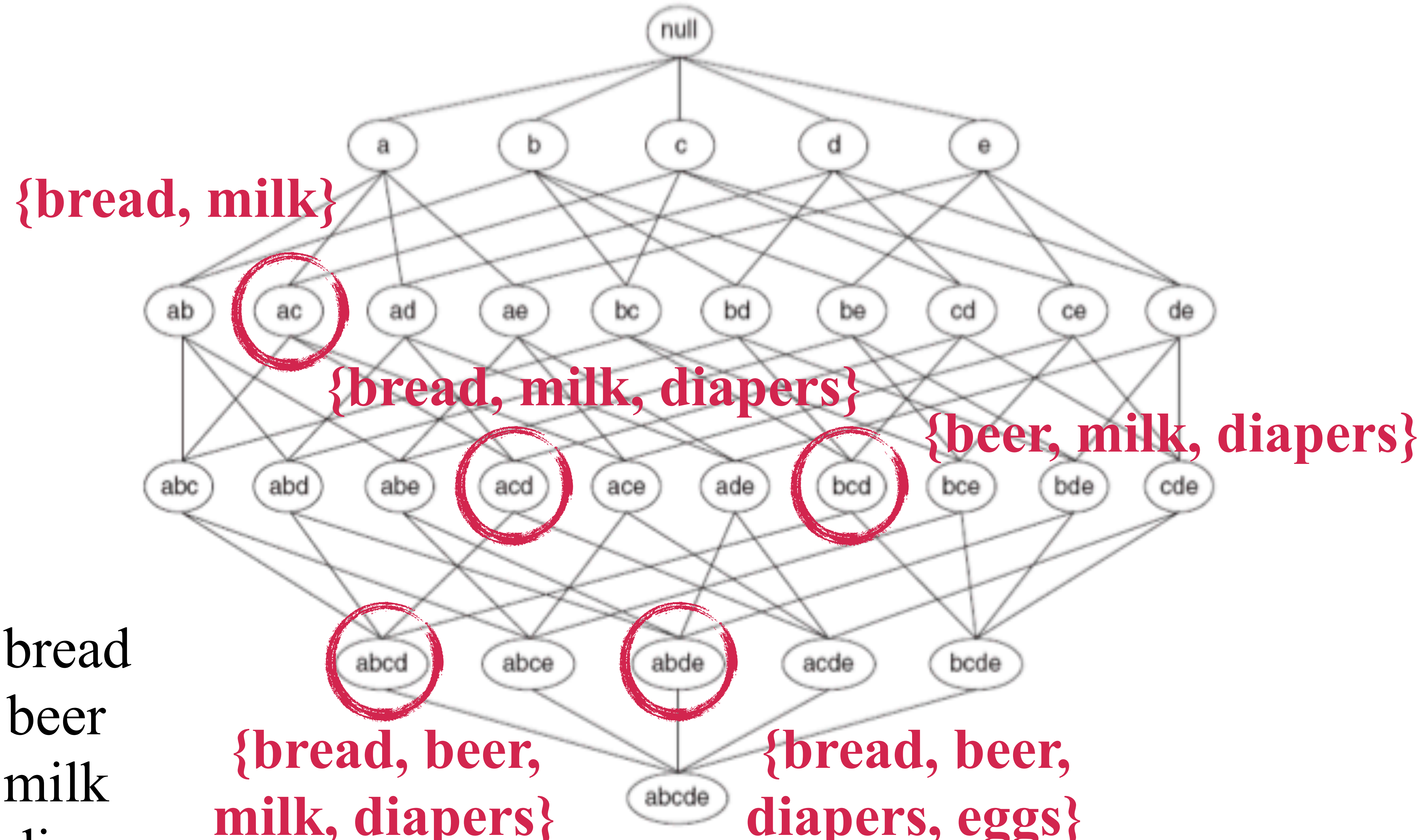
The market basket data

Items are: bread, milk, diapers, beer, and eggs
Transactions are: 1:{bread, milk}, 2:{bread, diapers, beer, eggs}, 3:{milk, diapers, beer}, 4:{bread, milk, diapers, beer}, and 5:{bread, milk, diapers}

Transaction IDs

TID	Bread	Milk	Diapers	Beer	Eggs
1	✔	✔			
2	✔		✔	✔	✔
3		✔	✔	✔	
4	✔	✔	✔	✔	
5	✔	✔	✔		

Transaction data as subsets



a: bread
b: beer
c: milk
d: diapers
e: eggs

2^n subsets of n items. Layer k has $\binom{n}{k}$ subsets.

Transaction data as binary matrix

TID	Bread	Milk	Diapers	Beer	Eggs
1	1	1	0	0	0
2	1	0	1	1	1
3	0	1	1	1	0
4	1	1	1	1	0
5	1	1	1	0	0

Any data that can be expressed as a binary matrix can be used.

Itemsets, support, and frequency

- An **itemset** is a set of items
  - A transaction t is an itemset with associated transaction ID, $t = (tid, I)$, where I is the set of items of the transaction
- A transaction $t = (tid, I)$ contains itemset X if $X \subseteq I$
- The **support** of itemset X in database D is the number of transactions in D that contain it:

$$supp(X, D) = |\{t \in D : t \text{ contains } X\}|$$

- The **frequency** of itemset X in database D is its support relative to the database size, $supp(X, D) / |D|$
- Itemset is **frequent** if its frequency is above user-defined threshold **minfreq** Mine these

Frequent itemset example

TID	Bread	Milk	Diapers	Beer	Eggs
1	1	1	0	0	0
2	1	0	1	1	1
3	0	1	1	1	0
4	1	1	1	1	0
5	1	1	1	0	0

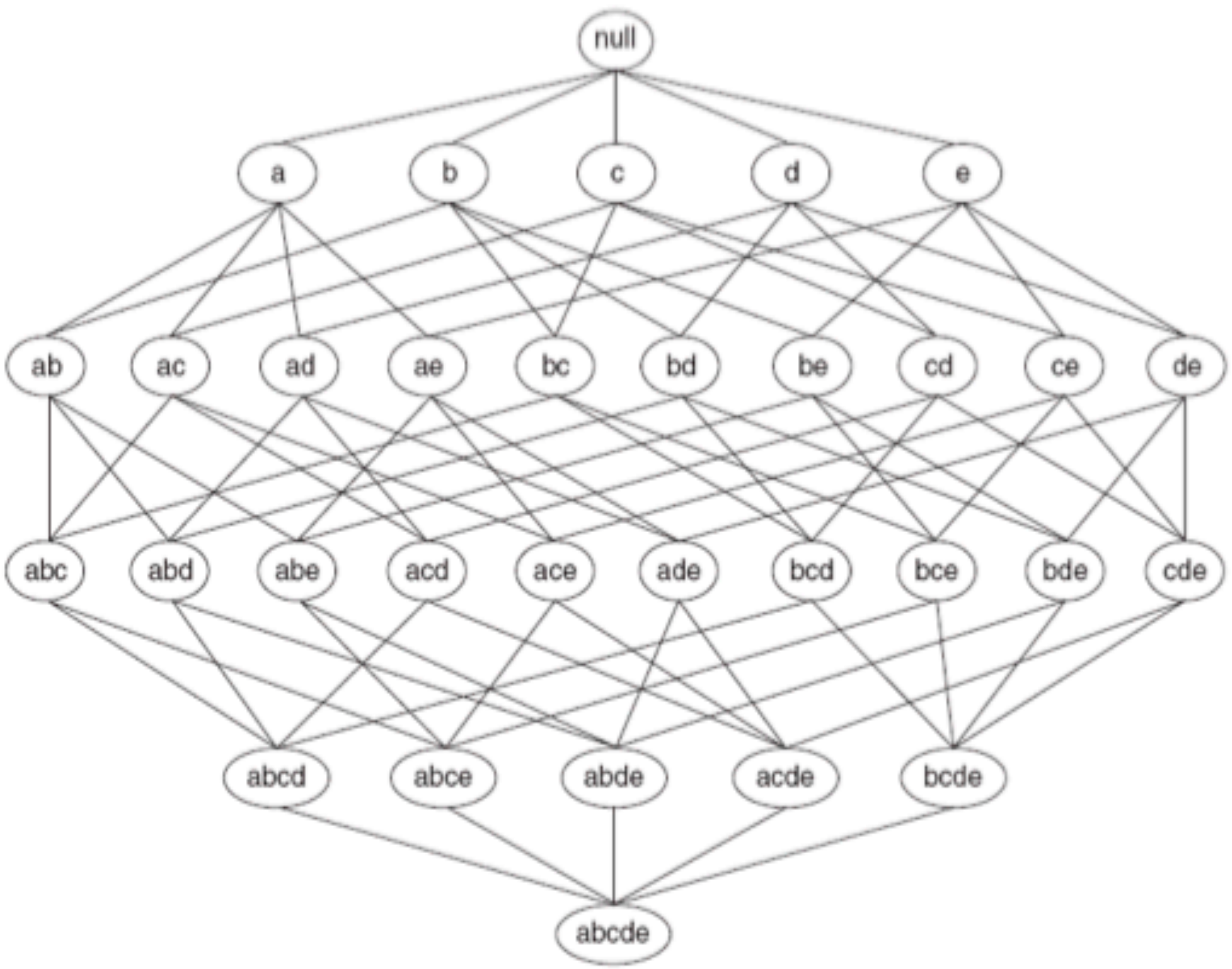
Itemset {Bread, Milk} has support 3 and frequency 3/5
Itemset {Bread, Milk, Eggs} has support and frequency 0
For **minfreq** = 1/2, frequent itemsets are:
{Bread}, {Milk}, {Diapers}, {Beer}, {Bread, Milk}, {Bread, Diapers}, {Milk, Diapers}, and {Diapers, Beer}

The Apriori Algorithm

- To find all the frequent itemsets we can just try all the possible itemsets
  - But there are $2^{|I|}$ itemsets ($|I|$ is the number of items)
- We can make this faster by reducing
  - the number of itemsets we consider
  - the number of transactions in the data
  - the number of comparisons of itemsets to transactions
- The Apriori algorithm reduces the number of itemsets we consider

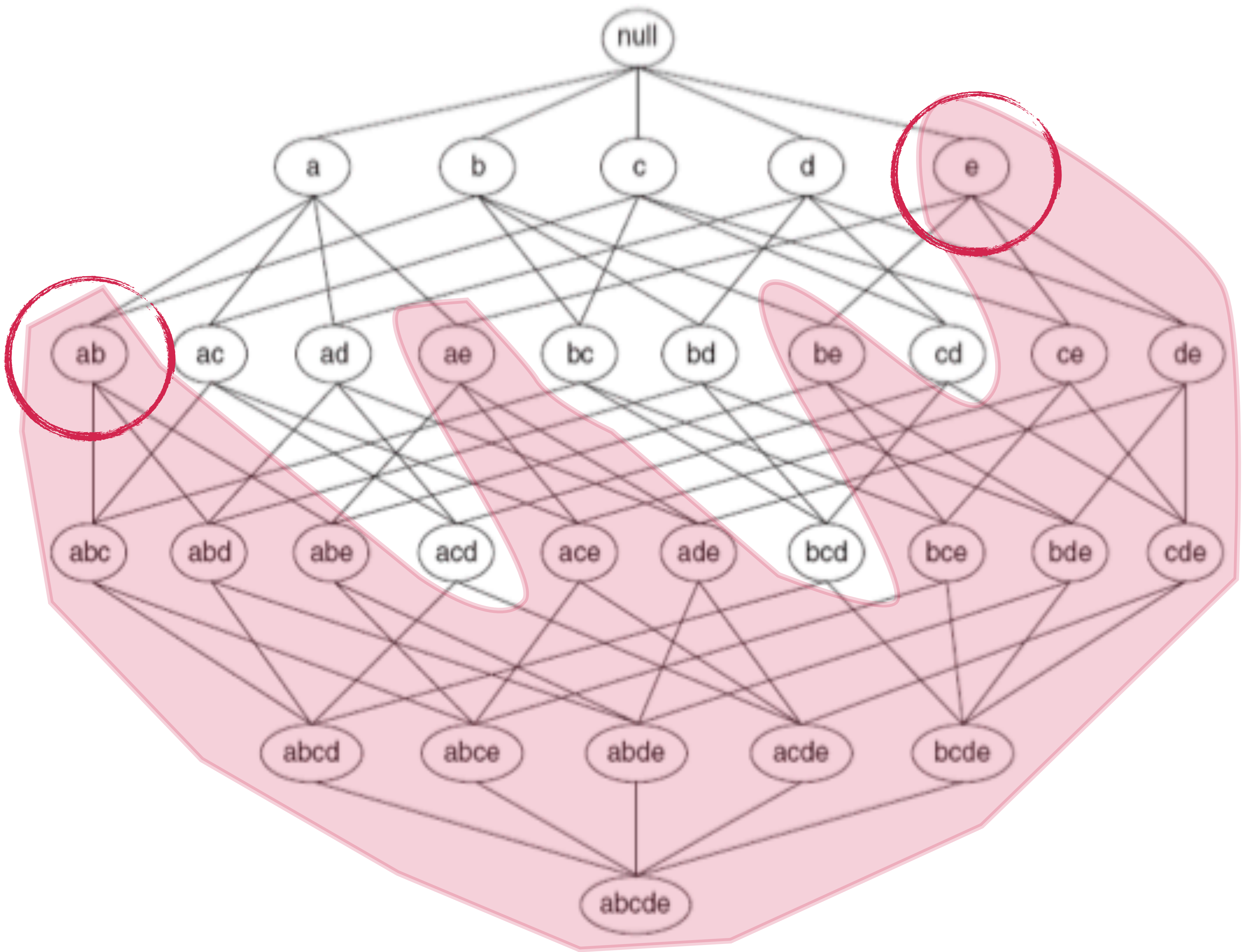
The Downwards Closedness Property

If X and Y are two itemsets such that $X \subset Y$, then $supp(Y) \leq supp(X)$.



Example of pruning itemsets

If {e} and {ab} are infrequent



Comments on Apriori

- The worst-case running time of Apriori is still $O(|I| \times |D| \times 2^{|I|})$
  - If all itemsets are frequent
- This can be improved to $O(|D| \times 2^{|I|})$ by storing the *tid*-lists of the itemsets together with them
  - The *Eclat* algorithm
  - Better I/O, as we don't have to query the data base for the support of each candidate itemset
- Third well-known method is the *FP-growth* algorithm
- In practice all these algorithms are very fast *unless the data is very dense or the threshold is too low*

The Flood of Itemsets

- Consider the following table:

tid	A	B	C	D	E	F	G	H
1	✔	✔	✔	✔	✔			
2		✔	✔	✔	✔	✔	✔	
3			✔	✔	✔	✔	✔	✔
4	✔	✔			✔	✔	✔	✔
5		✔	✔		✔	✔		✔
6	✔			✔	✔	✔		✔
7	✔	✔	✔	✔	✔	✔	✔	✔

- How many itemsets with minimum frequency of 1/7 it has?
- 255!
- Still 31 frequent itemsets with 50% minfreq

- ”Data mining is … to summarize the data”
  - Hardly a summarization!

Closed and maximal itemsets

- Let F be the set of all frequent itemsets (w.r.t. some **minfreq**) in data D
- Frequent itemset $X \in F$ is **maximal** if it does not have any frequent supersets
  - That is, for all $Y \supset X$, $Y \notin F$
- Frequent itemset $X \in F$ is **closed** if it has no superset with the same frequency
  - That is, for all $Y \supset X$, $supp(Y, D) < supp(X, D)$
    - It can't be that $supp(Y, D) > supp(X, D)$. Why?

Example of maximal frequent itemsets

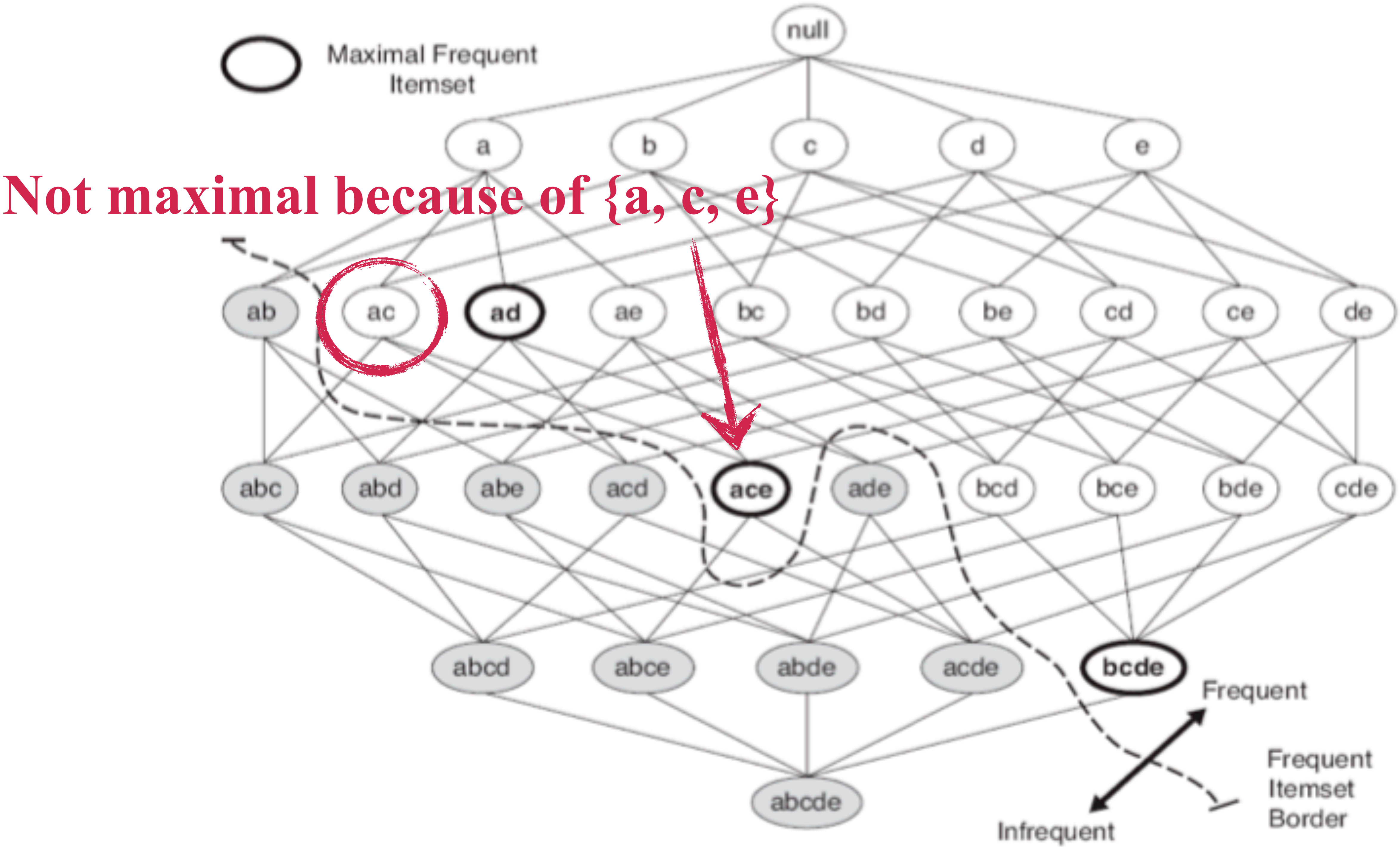
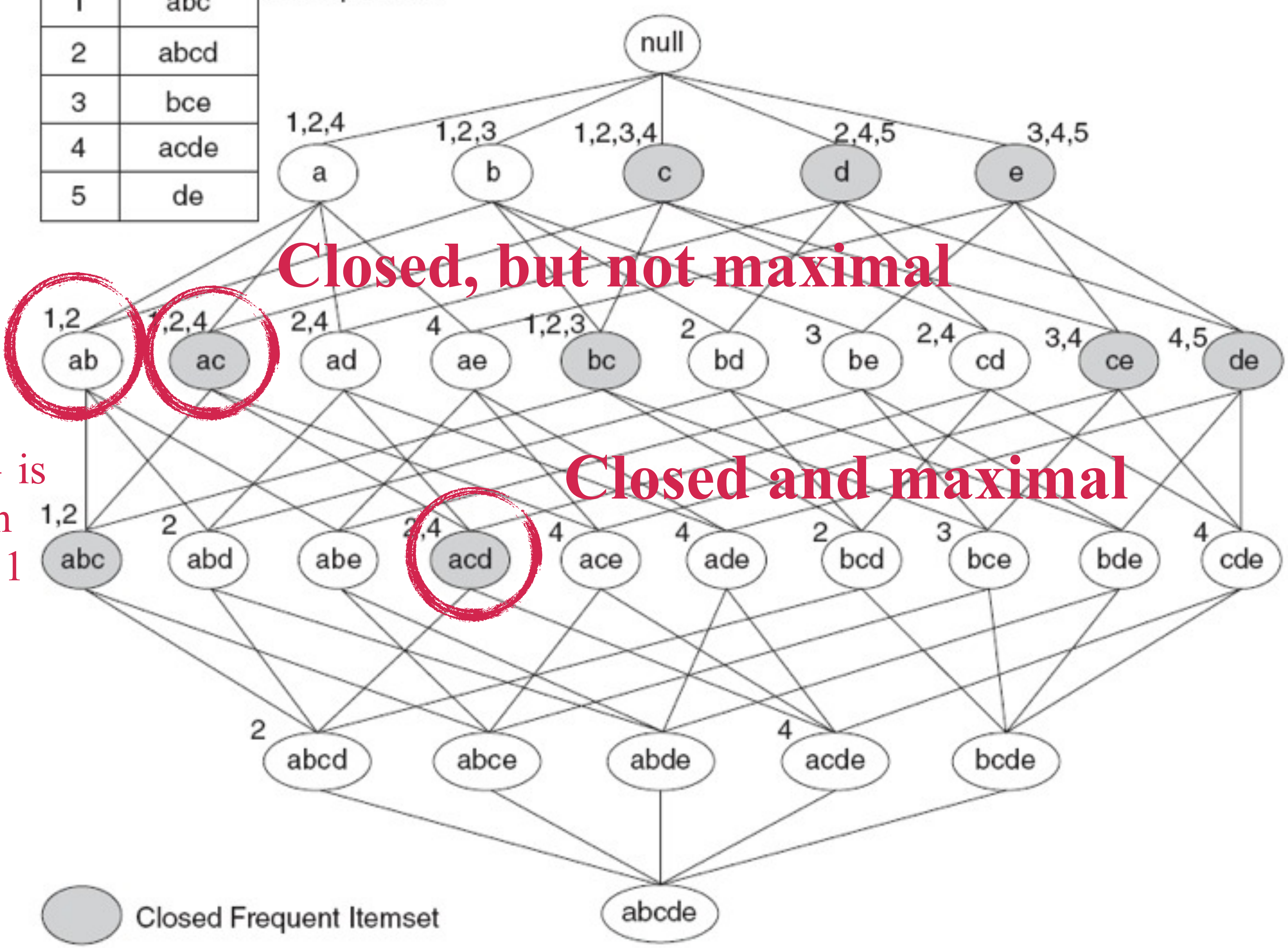


Figure 6.16. Maximal frequent itemset.

Example of closed frequent itemsets

TID	Items
1	abc
2	abcd
3	bce
4	acde
5	de



Itemset {a, b} is contained in transactions 1 and 2

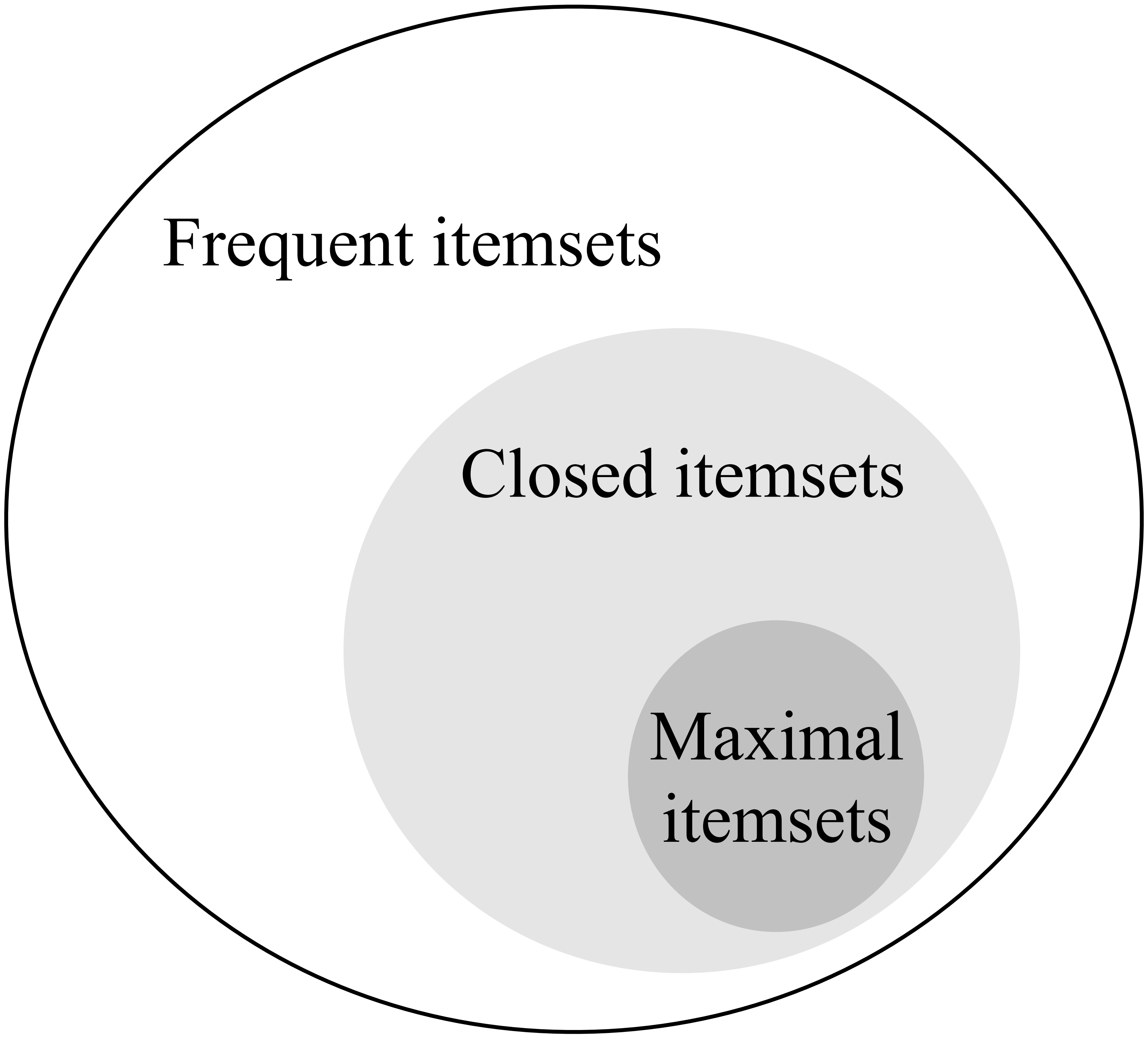
Figure 6.17. An example of the closed frequent itemsets (with minimum support count equal to 40%).

Closed & Maximal Itemsets

tid	A	B	C	D	E	F	G	H
1	✔	✔	✔	✔	✔			
2		✔	✔	✔	✔	✔	✔	
3			✔	✔	✔	✔	✔	✔
4	✔	✔			✔	✔	✔	✔
5		✔	✔		✔	✔		✔
6	✔			✔	✔	✔		✔
7	✔	✔	✔	✔	✔	✔	✔	✔

- 31 frequent itemsets with 50% minfreq
- 16 frequent *closed* itemsets with 50% minfreq
- 9 frequent *maximal* itemsets with 50% minfreq

Itemset taxonomy



Mining Maximal Itemsets

- The naïve approach:
  - Find all frequent itemsets and test each for maximality
  - When considering itemset X, if it is not a subset of existing maximal itemset Y, add it to set of *candidates* $\mathcal{M}$
    - If $\mathcal{M}$ has itemset Y s.t. $Y \subset X$, remove Y
  - Time complexity $O(|\mathcal{M}|)$
- Better approach (`GenMax`)
  - Search the itemset lattice in depth-first order
    - Only add X to $\mathcal{M}$ when sure X is maximal
    - Can prune whole branches when they're already contained in some maximal itemset

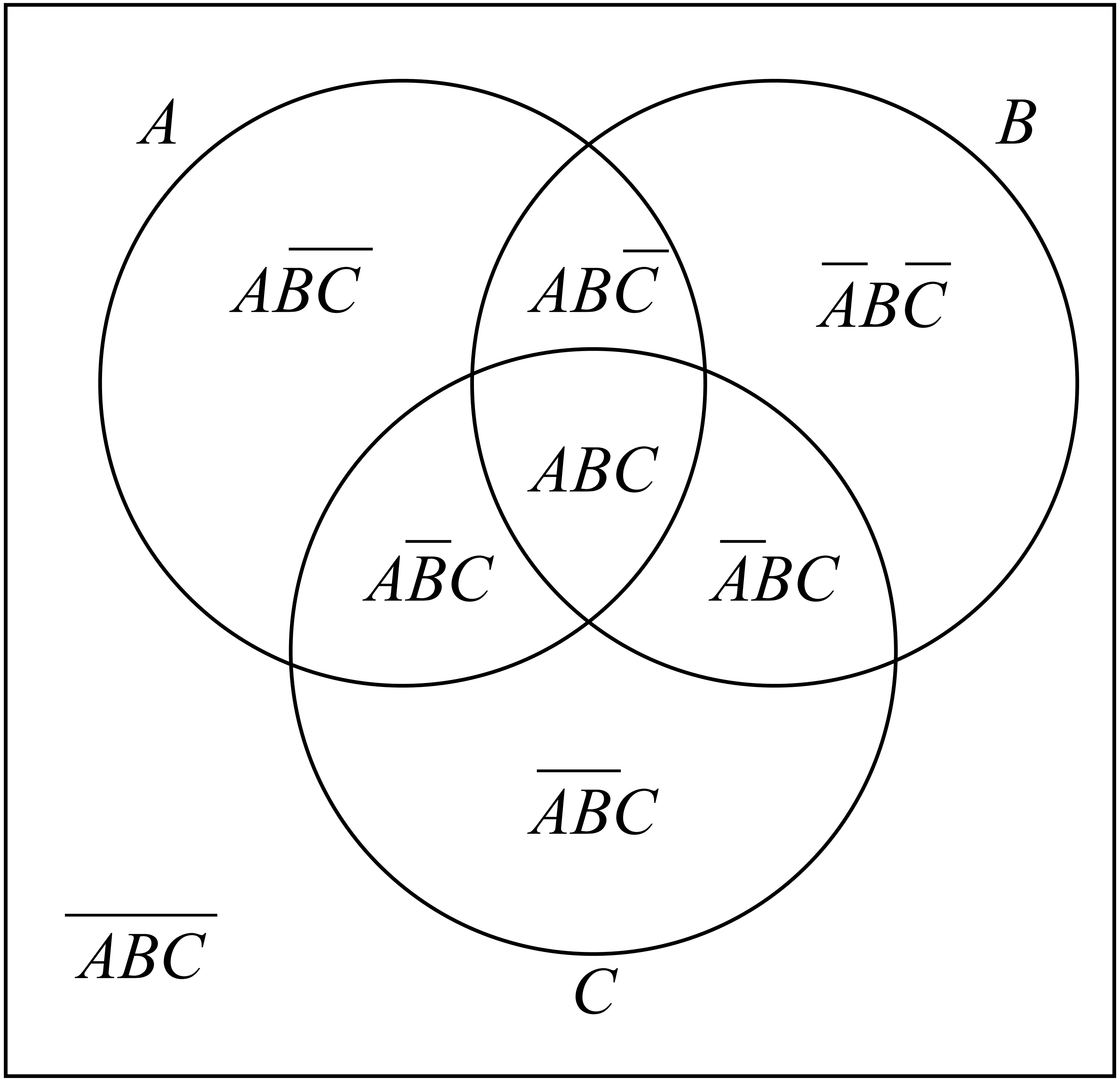
Mining Closed Itemsets

- Again, naïve approach is very expensive
- Three properties to reduce the itemsets to consider:
  1. If $\mathbf{t}(X_i) = \mathbf{t}(X_j)$, then $\mathbf{c}(X_i) = \mathbf{c}(X_j) = \mathbf{c}(X_i \cup X_j)$
     - $\mathbf{t}(X_i)$ = transactions of itemset X_i; $\mathbf{c}(X_i)$ the *closure* of X_i
     - We can replace every X_i with $X_i \cup Xj$ and prune away the branch under X_j
  2. If $\mathbf{t}(X_i) \subset \mathbf{t}(X_j)$, then $\mathbf{c}(X_i) \neq \mathbf{c}(X_j)$ but $\mathbf{c}(X_i) = \mathbf{c}(X_i \cup X_j)$
     - We can replace every occurrence of X_i with $X_i \cup X_j$, but cannot prune X_j
  3. If $\mathbf{t}(X_i) \neq \mathbf{t}(X_j)$, then $\mathbf{c}(X_i) \neq \mathbf{c}(X_j) \neq \mathbf{c}(X_i \cup X_j)$
     - There's nothing we can do
- The CHARM algorithm uses these properties

Non-Derivable Itemsets

- Let F be the set of all frequent itemsets. Itemset $X \in F$ is **non-derivable** if we cannot derive its support from its subsets.
  - We can derive the support of X from its subsets if, by knowing the supports of all of the subsets of X we can compute the support of X
- If X is derivable, it doesn't add any new information
  - Knowing just the non-derivable frequent itemsets, we can construct every frequent itemset
  - We only return itemsets that add new information on top of what we already knew

Generalized Itemsets



The Support of a Generalized Itemset

- A *generalized itemset* is an itemset of form $X\bar{Y}$
  - All items is X and no items in Y
- The *support* of a generalized itemset $X\bar{Y}$ is the number of transactions that contain all the items in X, but *no* items in Y
- To compute the support of a generalized itemset $A\overline{BC}$, we can
  - Take the support of A
  - Remove the supports of AB and AC
  - Add the support of ABC that was removed twice
  - $supp(A\overline{BC}) = supp(A) - supp(AB) - supp(AC) + supp(ABC)$

The Inclusion-Exclusion Principle

- Let $X\bar{Y}$ be a generalized itemset and let $I = X \cup Y$
- Now $supp(X\bar{Y})$ can be expressed as a combination of supports of supersets $J \supseteq X$ such that $J \subseteq I$ using the **inclusion-exclusion principle**

$$supp(X\bar{Y}) = \sum_{X \subseteq J \subseteq I} (-1)^{|J \setminus X|} supp(J)$$

  – Example:

$$\begin{aligned} supp(\overline{ABC}) = {} & supp(\emptyset) \\ & - supp(A) - supp(B) - supp(C) \\ & + supp(AB) + supp(AC) + supp(BC) \\ & - supp(ABC) \end{aligned}$$

Support Bounds

- The inclusion-exclusion formula gives us bounds for the supports of itemsets in $X \cup Y$ that are supersets of X
  - All supports are non-negative!
  - $supp(A\overline{BC}) = supp(A) - supp(AB) - supp(AC) + supp(ABC) \geq 0$ implies $supp(ABC) \geq -supp(A) + supp(AB) + supp(AC)$
    - This is a lower bound, but we can also get upper bounds
- In general the bounds for itemset I w.r.t. $X \subset I$:
  - If $|I \setminus X|$ is odd: $supp(I) \leq \sum_{X \subseteq J \subseteq I} (-1)^{|I \setminus J|+1} supp(J)$
  - If $|I \setminus X|$ is even: $supp(I) \geq \sum_{X \subseteq J \subseteq I} (-1)^{|I \setminus J|+1} supp(J)$

Deriving the Support

- Given the formula for the bounds, we can define
  - the *least upper bound lub(I)* and
  - the *greatest lower bound glb(I)* for itemset *I*
- We know that $supp(I) \in [glb(I), lub(I)]$
- If $glb(I) = lub(I)$, then we can compute $supp(I)$ by just knowing its subsets' supports
  - Hence, *I* is derivable
- Otherwise *I* is non-derivable

Local and Global Data Mining

- Frequent itemset mining is *local*
  - Each itemset is evaluated on its own, irrespective of other itemsets
- Purely local evaluation tends to yield to explosion of patterns
- In *global* data mining the patterns are evaluated given the other patterns we know and the data as a whole
  - E.g. clustering
  - Closed, maximal, and non-derivable itemsets move from local towards global, but don't care about the data
- Next two lectures: more global take on pattern mining