CSCI-620

Itemset Mining

“Progress in bar-code technology has made it possible for retail organizations to collect and store massive amounts of sales data, referred to as the basket data. A record in such data typically consists of the transaction date and the items bought in the transaction. Successful organizations view such databases as important pieces of the marketing infrastructure. They are interested in instituting information-driven marketing processes, managed by database technology, that enable marketers to develop and implement customized marketing programs and strategies.”

Bar codes

Beer and diapers

It goes back to 1992, Osco:

"...90 days of point-of-sale data from Osco Drug stores -1.2 million baskets...
...between 5pm and 7pm, customers tended to co-purchase beer and diapers
....we have a correlation between beer, diapers and time..."

	Milk	Bread	Butter	Beer	Diapers
0	Yes	Yes	Yes	No	No
1	No	Yes	Yes	No	No
2	Yes	Yes	No	No	Yes
3	No	Yes	Yes	Yes	No
4	Yes	Yes	Yes	Yes	Yes
5	Yes	Yes	Yes	No	Yes

{Bread, Butter} → Milk (3/5 confidence)

Beer → Diapers (1/2 confidence)

Association

	Milk	Bread	Butter	Beer	Diapers
0	Yes	Yes	Yes	No	No
1	No	Yes	Yes	No	No
2	Yes	Yes	No	No	Yes
3	No	Yes	Yes	Yes	No
4	Yes	Yes	Yes	Yes	Yes
5	Yes	Yes	Yes	No	Yes

Counting frequent items

Items	Freq.
Milk	4
Bread	6
Butter	5
[redacted]	[redacted]
Diapers	3

Items	Freq.
Milk, Bread	4
Milk, Butter	3
Milk, Diapers	3
Bread, Butter	5
Bread, Diapers	3
[redacted]	[redacted]

Items	Freq.
Milk, Bread, Butter	3
Milk, Bread, Diapers	3
[redacted]	[redacted]
[redacted]	[redacted]

Counting frequent items

Why do we care?

- Increase co-purchases
- Cross-promotion
- Price optimization
- Inventory management
- Refine marketing

Correlation vs Causation

- Correlation - two values which tend to have similar or opposite trends
- Causation - a change in one variable results in a change in another
- Correlation != causation

Correlation vs Causation

1. Grocery, ~1994, No correlation
2. Drug store, ~1995, Correlation
3. Drug store, 1997, Very high correlation
4. Drug store, 1997, No correlation
5. Grocery, 2000, Weak correlation
6. Multiple, 2013, Correlation

What can we do?

- What values tend to rise (or fall) together?
- What events tend to co-occur?

$I = \{i_1, i_2, \ldots, i_m\}$ is a set of items, e.g., Beer or Diapers

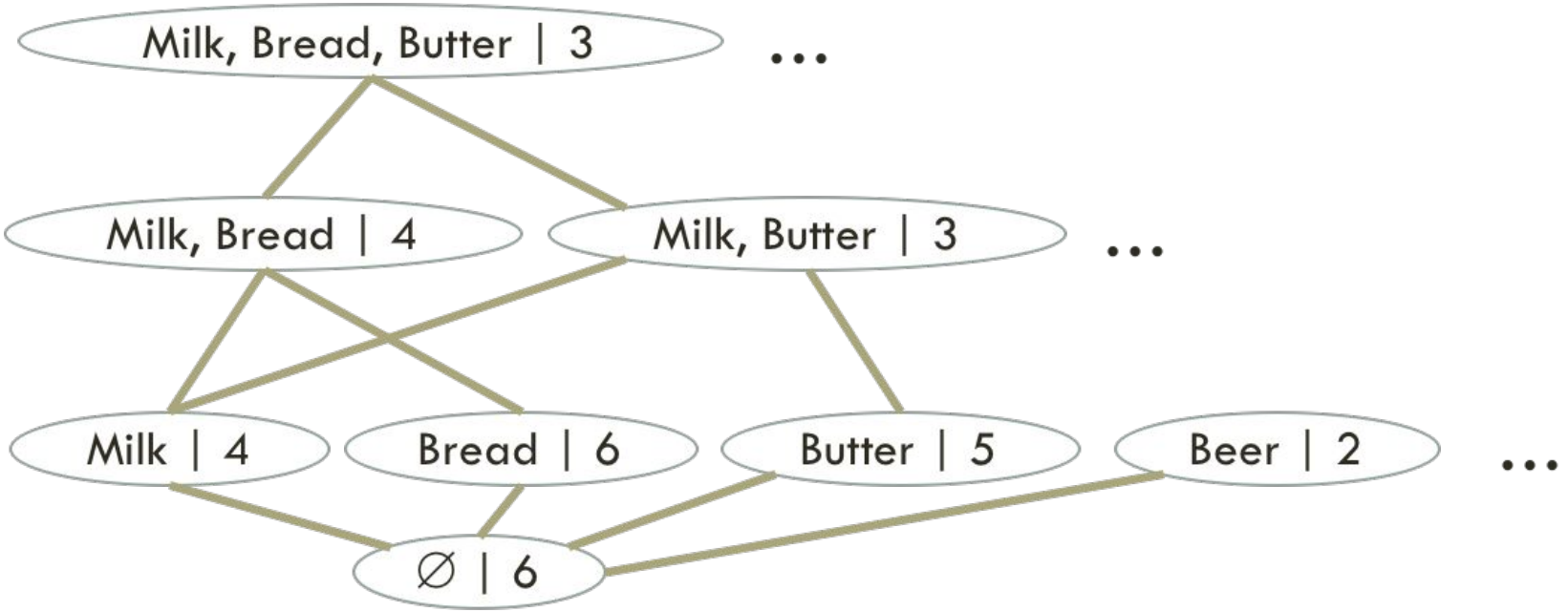
D is a set of *transactions* where each transaction T is a set of items such that $T \subseteq I$; each T has associated a unique identifier called TID

A *frequent itemset* $F \subseteq I$ is a set of items which occurs in some number of transactions in T greater than a parameter which we will call the *minimum support*

Formal statement

Frequent itemsets

- **Support** - the number of transactions including two items
- We define a *minimum support* required for rules to be considered interesting
- Starting with single item sets, we build new sets at each level if they continue to meet the minimum support


Milk, Bread, Butter | 3
...
Milk, Bread | 4
Milk, Butter | 3
...
Milk | 4
Bread | 6
Butter | 5
Beer | 2
...
∅ | 6

Lattices

Notation

- ***k*-itemset** - an itemset of size *k* with elements sorted lexicographically
- L_k - set of *k*-itemsets with minimum support containing items and a count
- C_k - set of candidate *k*-itemsets, each of which contains the items and a count

$L_1 = \{\text{large 1-itemsets}\};$
for ($k = 2$; $L_{k-1} \neq \emptyset$; k++) **do begin**
  $C_k = \text{apriori-gen}(L_{k-1})$; // New candidates
  forall transactions $t \in \mathcal{D}$ **do begin**
    $C_t = \text{subset}(C_k, t)$; // Candidates contained in t
    forall candidates $c \in C_t$ **do**
      $c.\text{count}$++;
  end
  $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$
end
Answer $= \bigcup_k L_k$;

Apriori algorithm

insert into C_k
select $p.\text{item}_1, p.\text{item}_2, \ldots, p.\text{item}_{k-1}, q.\text{item}_{k-1}$
from $L_{k-1}\ p, L_{k-1}\ q$
where $p.\text{item}_1 = q.\text{item}_1, \ldots, p.\text{item}_{k-2} = q.\text{item}_{k-2},$
$p.\text{item}_{k-1} < q.\text{item}_{k-1};$

Apriori-gen (join)

forall itemsets $c \in C_k$ **do**
  forall $(k-1)$-subsets s of c **do**
    if $(s \notin L_{k-1})$ **then**
      delete c from C_k;

Apriori-gen (prune)

Transaction	Items
1	Milk, Bread, Butter
2	Bread, Cheese
3	Bread, Jam
4	Milk, Bread, Cheese
5	Milk, Jam
6	Bread, Jam
7	Milk, Jam
8	Milk, Bread, Jam, Butter
9	Milk, Bread, Jam

Minimum Support
2

Example

Items	Support
{Milk}	6
{Bread}	7
{Butter}	2
{Cheese}	2
{Jam}	6

1-itemsets (L_1)

Items	Support
{Milk}	6
{Bread}	7
{Butter}	2
{Cheese}	2
{Jam}	6

{Bread, Milk}
{Butter, Milk}
{Cheese, Milk}
{Jam, Milk}
{Bread, Butter}
{Bread, Cheese}
{Bread, Jam}
{Butter, Cheese}
{Butter, Jam}
{Cheese, Jam}

2-itemsets (C_2)

Items	Support
{Bread, Milk}	4
{Butter, Milk}	2
~~{Cheese, Milk}~~	~~1~~
{Jam, Milk}	2
{Bread, Butter}	2
{Bread, Cheese}	2
{Bread, Jam}	4
~~{Butter, Cheese}~~	~~0~~
~~{Butter, Jam}~~	~~1~~
~~{Cheese, Jam}~~	~~0~~

Items	Support
{Bread, Milk}	4
{Butter, Milk}	2
{Jam, Milk}	2
{Bread, Butter}	2
{Bread, Cheese}	2
{Bread, Jam}	4

2-itemsets (L_2)

Items	Support
{Bread, Milk}	4
{Butter, Milk}	2
{Jam, Milk}	2
{Bread, Butter}	2
{Bread, Cheese}	2
{Bread, Jam}	4

{Bread, Butter, Milk}
{Bread, Jam, Milk}
~~{Bread, Cheese, Milk}~~
{Bread, Butter, Milk}
~~{Bread, Butter, Cheese}~~

3-itemsets (C_3)

Items	Support
{Bread, Butter, Milk}	2
{Bread, Jam, Milk}	2
~~{Bread, Cheese, Milk}~~	~~1~~
~~{Butter, Bread, Milk}~~	~~1~~
~~{Bread, Butter, Cheese}~~	~~0~~

Items	Support
{Bread, Butter, Milk}	2
{Bread, Jam, Milk}	2

No frequent 4-itemsets!

3-itemsets (L_3)

PCY Algorithm

- Improves on the memory usage as compared to the apriori algorithm
- On the first pass, hash all pairs of items and add an one to a count at that entry
- On the second pass, we only need to check pairs whose bucket meets our minimum support

Items	Support	Bucket
{Bread, Milk}	4	0
{Butter, Milk}	2	0
{Cheese, Milk}	1	1
{Jam, Milk}	2	1
{Bread, Butter}	2	2
{Bread, Cheese}	2	2
{Bread, Jam}	4	3
{Butter, Cheese}	0	3
{Butter, Jam}	1	4
{Cheese, Jam}	0	4

Bucket	Count
0	6
1	3
2	4
3	4
4	1

2-itemsets (C_2)

Items	Support	Bucket
{Bread, Milk}	4	0
{Butter, Milk}	2	0
{Cheese, Milk}	1	1
{Jam, Milk}	2	1
{Bread, Butter}	2	2
{Bread, Cheese}	2	2
{Bread, Jam}	4	3
{Butter, Cheese}	0	3
~~{Butter, Jam}~~	~~1~~	~~4~~
~~{Cheese, Jam}~~	~~0~~	~~4~~

Bucket	Count
0	6
1	3
2	4
3	4
~~4~~	~~1~~

2-itemsets (C_2)

Items	Support	Bucket
{Bread, Milk}	4	0
{Butter, Milk}	2	0
~~{Cheese, Milk}~~	~~1~~	~~1~~
{Jam, Milk}	2	1
{Bread, Butter}	2	2
{Bread, Cheese}	2	2
{Bread, Jam}	4	3
~~{Butter, Cheese}~~	~~0~~	~~3~~

Bucket	Count
0	6
1	3
2	4
3	4
4	1

2-itemsets (L_2)

Items	Support
~~{Milk}~~	~~6~~
~~{Bread}~~	~~7~~
~~{Butter}~~	~~2~~
~~{Cheese}~~	~~2~~
~~{Jam}~~	~~6~~

Items	Support
~~{Bread, Milk}~~	~~4~~
~~{Butter, Milk}~~	~~2~~
{Jam, Milk}	2
~~{Bread, Butter}~~	~~2~~
{Bread, Cheese}	2
{Bread, Jam}	4

Items	Support
{Bread, Butter, Milk}	2

Maximal frequent itemsets

Items	Support
{Jam, Milk}	2
{Bread, Cheese}	2
{Bread, Jam}	4

Items	Support
{Bread, Butter, Milk}	2

Maximal frequent itemsets

Association rule mining

- Association rules are of the form $X \rightarrow Y$
- This suggests if X is in a transaction, then Y is likely to occur in the same transaction
- Rule mining algorithms use a *strength* and/or *interestingness* metric to suggest possible rules, confirmed via support

$I = \{i_1, i_2, \dots, i_m\}$ is a set of items, e.g., Beer or Diapers

D is a set of *transactions* where each transaction T is a set of items such that $T \subseteq I$; each T has associated a unique identifier called TID

$X \rightarrow Y$ is an *association rule* where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$

$X \rightarrow Y$ holds in D with *confidence* c if at least c% of transactions in D containing X also contain Y

$X \rightarrow Y$ has *support* s if s% of transactions in D contain $X \cup Y$

Formal statement

Given a set of transactions D, we wish to generate all association rules X $\rightarrow$Y that have support and confidence greater than the user-specified minimum support (called minsup) and minimum confidence (called minconf), respectively

Formal statement

$$confidence(X \rightarrow Y) = p(Y|X) = \frac{support(X \cup Y)}{support(X)}$$

$$lift(X \rightarrow Y) = \frac{support(X \cup Y)}{support(X) \times support(Y)}$$

(here, support is fraction of transactions, not a count)

Confidence and lift

Rule generation

- For each frequent itemset ($k > 1$), consider possible association rules

- {A, B, C, D}

A → BCD	AB → CD	ABC → D
B → ACD	BC → AD	BCD → A
C → ABD	AC → BD	ABD → C
D → ABC	AD → BC	ACD → B
	BD → AC	
	CD → AB	

Transaction	Items
1	Milk, Bread, Butter
2	Bread, Cheese
3	Bread, Jam
4	Milk, Bread, Cheese
5	Milk, Jam
6	Bread, Jam
7	Milk, Jam
8	Milk, Bread, Jam, Butter
9	Milk, Bread, Jam

{Bread, Milk}

Example

Milk → Bread

support(Milk) = 6/9

support(Bread) = 7/9

support(Milk ∪ Bread) = 4/9

confidence(Milk → Bread) = (4/9)/(6/9) = 6/9

lift(Milk → Bread) = (4/9) / (6/9 * 7/9) = 6/7

Example

Rule generation

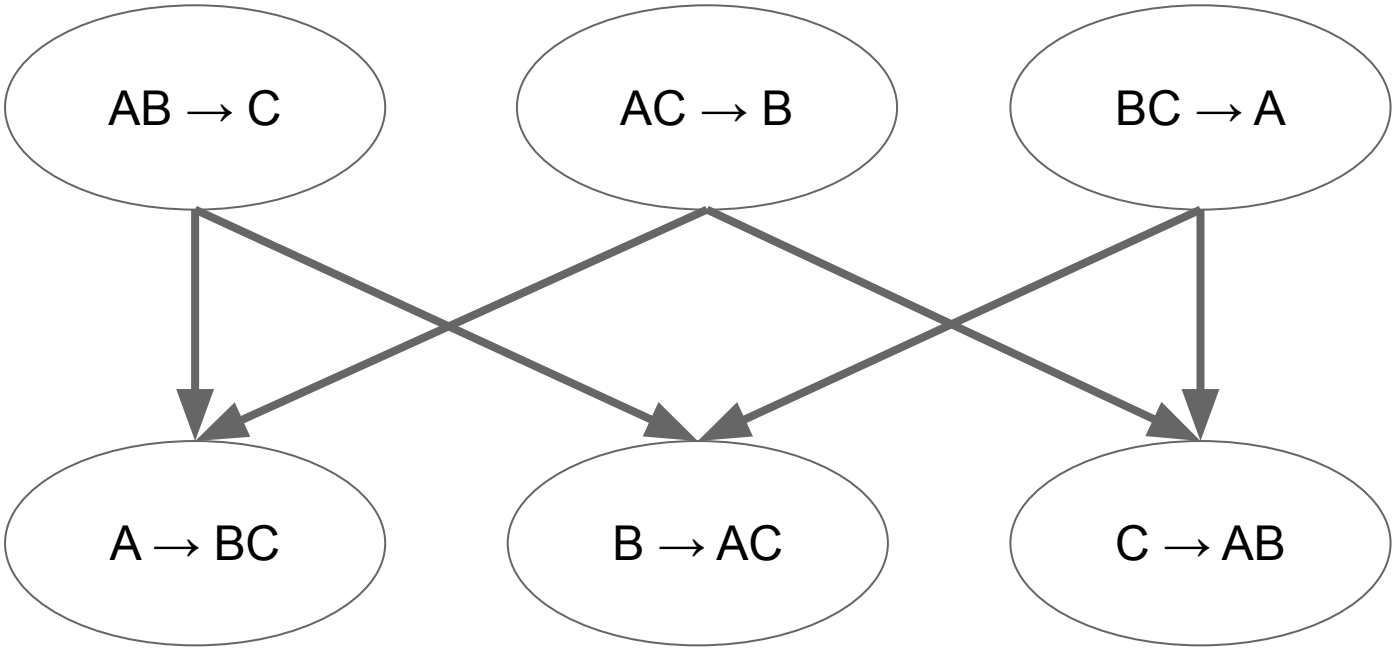
- For each frequent itemset ($k > 1$), consider possible association rules

- {A, B, C, D}

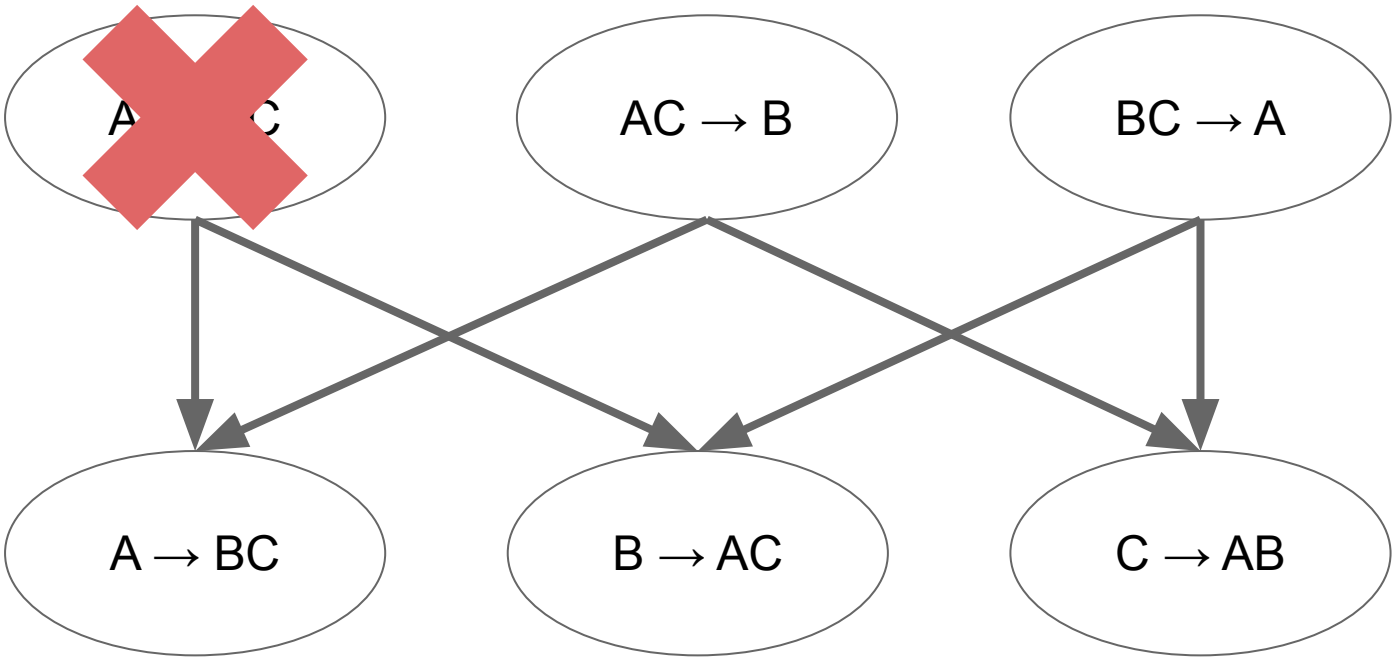
$A \rightarrow BCD$
$B \rightarrow ACD$
$C \rightarrow ABD$
$D \rightarrow ABC$

$AB \rightarrow CD$
$BC \rightarrow AD$
$AC \rightarrow BD$
$AD \rightarrow BC$
$BD \rightarrow AC$
$CD \rightarrow AB$

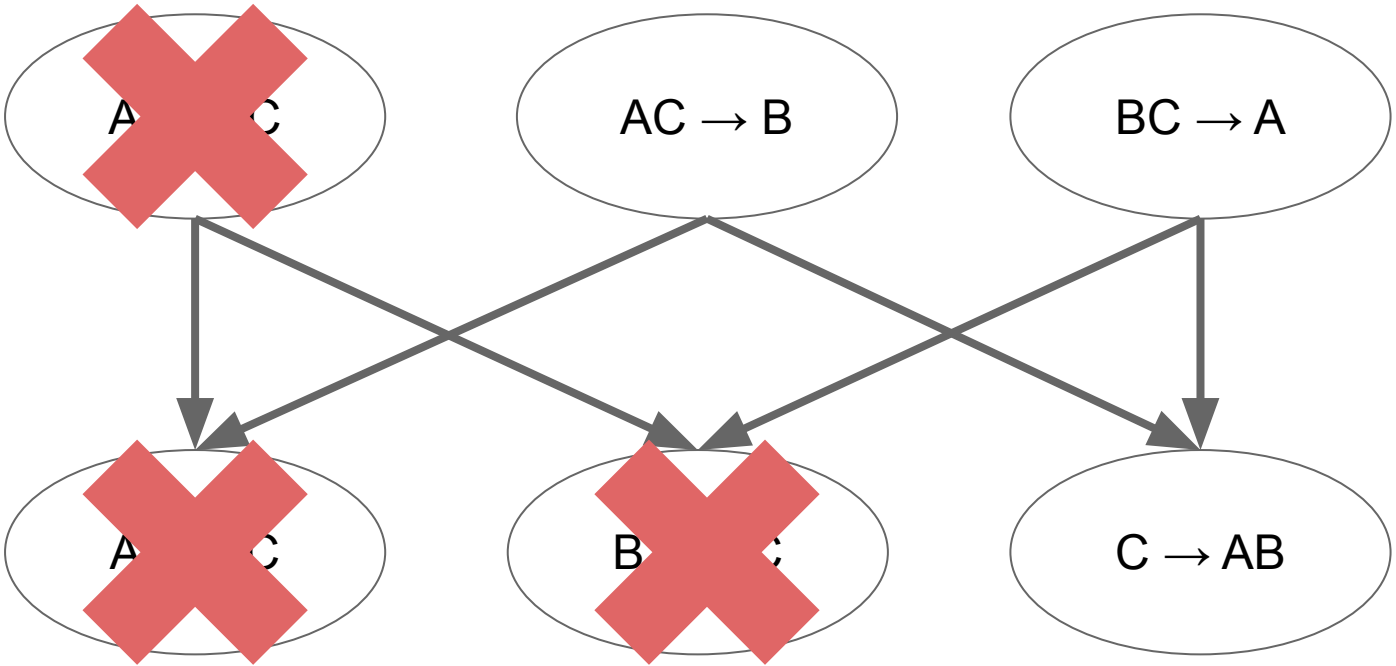
$ABC \rightarrow D$
$BCD \rightarrow A$
$ABD \rightarrow C$
$ACD \rightarrow B$



Rule lattice

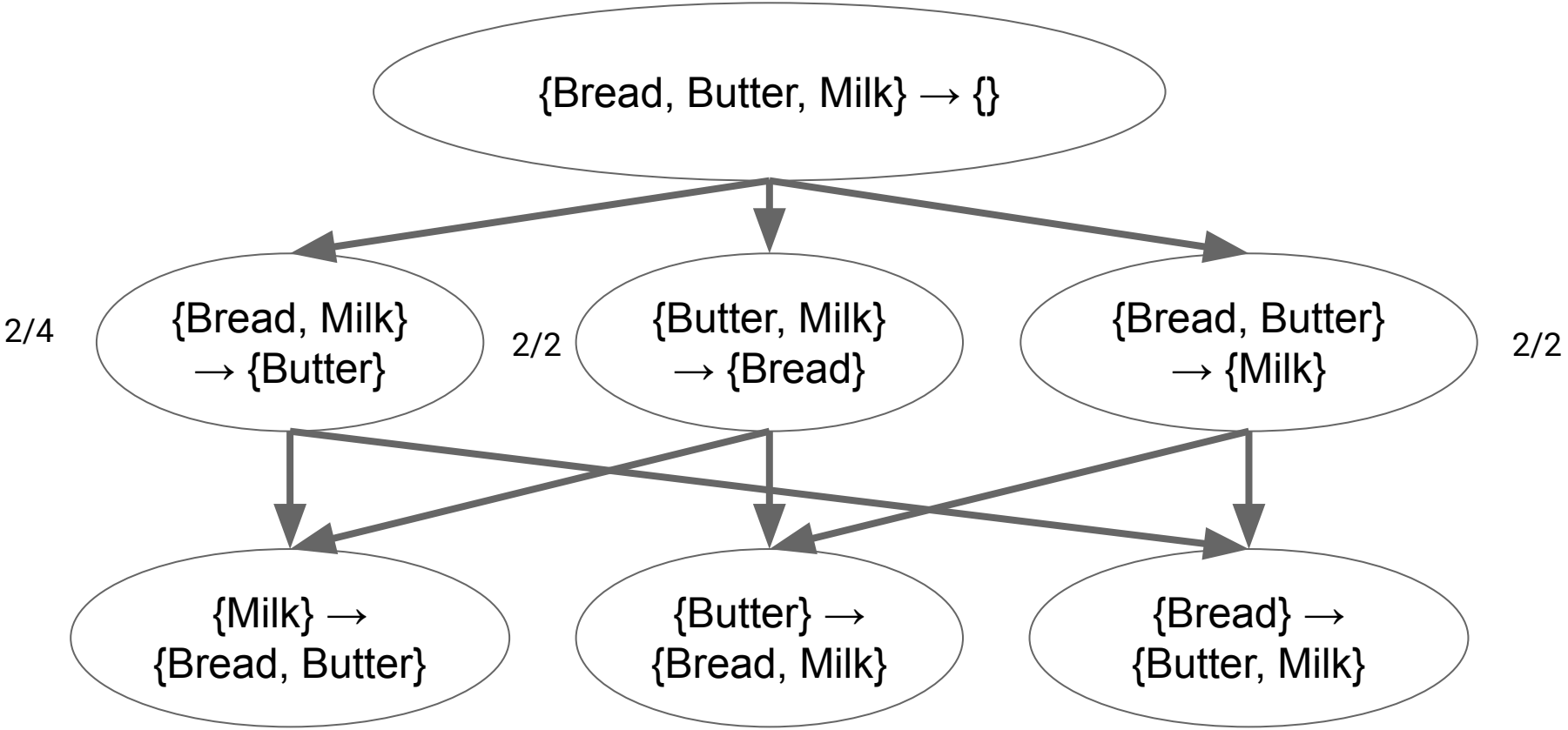


Rule lattice

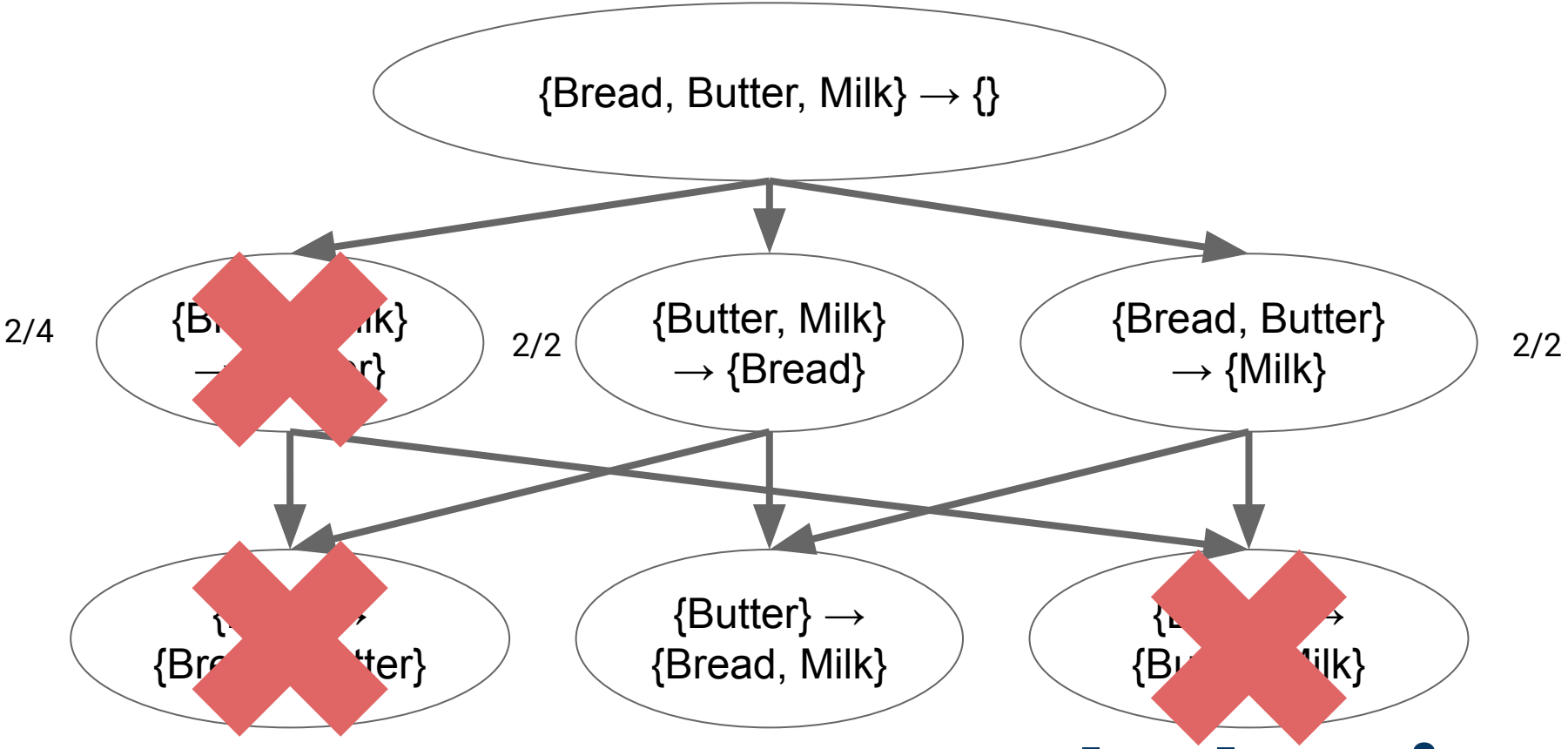


With less evidence (items on the left),
we cannot predict more items on the right!

Rule lattice



Rule lattice



Rule lattice