



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

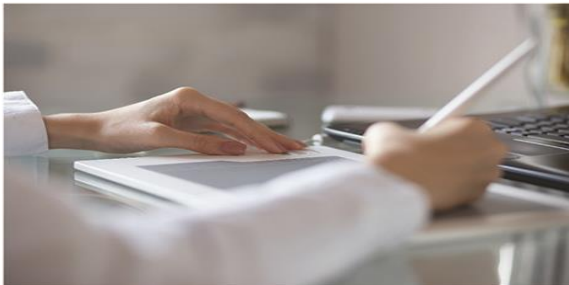
EDUCACIÓN
PROFESIONAL

Diplomado en Big Data y Ciencias de Datos

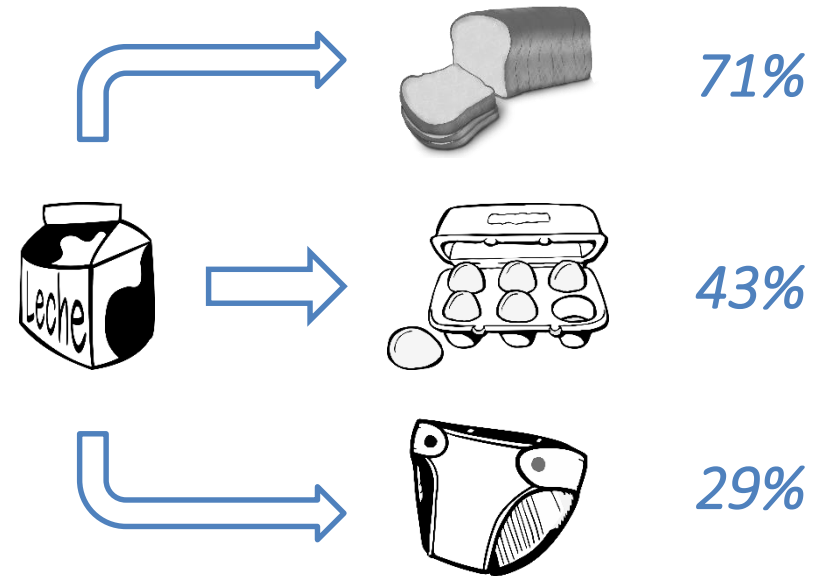
Minería de Datos Reglas de Asociación

Educación Profesional - Escuela de Ingeniería UC

Sebastián Raveau



¿Qué son las reglas de asociación?



De las transacciones que incluyeron leche:

- 71% incluyó pan
- 43% incluyó huevos
- 29% incluyó pañales

¿Para qué sirven las reglas de asociación?

La minería de asociación implica el uso de modelos de *machine learning* para encontrar patrones o coocurrencias dentro de una base de datos de transacciones

Busca identificar asociaciones del tipo “si a , entonces b ”, llamadas reglas de asociación, que poseen dos componentes: un antecedente (si) y un consecuente (entonces)

Un antecedente es un ítem presente en los datos; un consecuente es otro ítem presente en los datos junto al antecedente

Algunas aplicaciones de reglas de asociación

Ordenamiento de productos

Promoción de productos

Patrones y recomendaciones

Segmentación y ofertas

Algoritmos para obtener reglas de asociación

Son dos los algoritmos más utilizados:

Apriori (Agrawal y Srikant, 1994)

FP-Growth (Han, Pei y Yin, 2000)

Definiciones Preliminares

Itemset

Un itemset es un conjunto de uno o más ítems

Ejemplo: { Leche , Pan , Pañales }

La frecuencia de un itemset está dada por la cantidad de veces que aparece dentro de un conjunto de transacciones

Transacciones

Sean:

$I = \{ i_1, i_2, \dots, i_n \}$ el conjunto de todos los ítems

$T = \{ t_1, t_2, \dots, t_m \}$ el conjunto de todas las transacciones

Cada transacción en T posee una ID único y contiene un subconjunto de ítems de I

Ejemplo de transacciones

Transacción	Itemset
1	{ Pan , Leche }
2	{ Pan , Pañales , Cerveza , Huevos }
3	{ Leche , Pañales , Cerveza , Café }
4	{ Pan , Leche , Pañales , Cerveza }
5	{ Pan , Leche , Pañales , Café }
⋮	⋮

Ítems = { Pan , Leche , Pañales , Cerveza , Huevos , ... }

Transacción	Itemset
1	{ Pan }
1	{ Leche }
2	{ Pan }
2	{ Pañales }
2	{ Cerveza }
2	{ Huevos }
3	{ Leche }
3	{ Pañales }
3	{ Cerveza }
3	{ Café }
⋮	⋮

Regla de asociación

Es una expresión de la forma $X \Rightarrow Y$, donde X e Y son itemsets

Ejemplo: { Leche , Pañales } \Rightarrow { Cerveza }

 antecedente consecuente

Una regla de asociación se puede interpretar de la forma “si alguien compra leche y pañales, es posible que también compre cerveza”

Se obtienen a partir de observaciones de transacciones, y requieren de ciertos umbrales de consideración (e.g. qué significa “es posible” en la frase anterior)

Indicadores de Rendimiento

Soporte

El soporte de un itemset X corresponde a la frecuencia de transacciones a las que pertenece

$$s(X) = \frac{\# \text{ de transacciones que contienen a } X}{\# \text{ de transacciones}}$$

En general nos interesará obtener reglas de asociación entre itemsets con alto soporte

Ejemplo de Soporte

Transacción	Itemset
1	{ A , B }
2	{ A , C , D }
3	{ B , C , D , E }
4	{ A , B , C , D }
5	{ A , B , C , E }

Itemset	Soporte
{ A }	0,8
{ B }	0,8
{ C }	0,8
{ D }	0,6
{ E }	0,4
{ A , B }	0,6
{ A , D }	0,4
{ A , E }	0,2
{ A , B , E }	0,2
{ A , D , E }	0

Soporte

Sea Y un itemset que pertenece a otro itemset X más grande

$$Y \subseteq X \quad \rightarrow \quad S(X) \leq S(Y)$$

Un indicador relevante para construir reglas de asociación es analizar cuánto cambia (y en general disminuye) el soporte de un itemset al aumentar su tamaño

Confianza

La confianza de una regla de asociación $X \Rightarrow Y$ indica cuán frecuentemente es cierta

Corresponde a la cantidad de transacciones que contienen X que también contienen Y

$$C(X \Rightarrow Y) = \frac{s(X \cup Y)}{s(X)}$$

Ejemplo de Confianza

Transacción	Itemset
1	{ A , B }
2	{ A , C , D }
3	{ B , C , D , E }
4	{ A , B , C , D }
5	{ A , B , C , E }

$$s(\{ A , C \}) = 0,6$$

$$s(\{ A , C , D \}) = 0,4$$

$$c(\{ A , C \} \Rightarrow \{ D \}) = \frac{0,4}{0,6} = 0,\bar{6}$$

Un 66,7% de las veces que se compró A y C, también se compró D

Confianza

Supongamos que tenemos una regla de asociación $X \Rightarrow Y$ con una confianza de 80%

$$C(X \Rightarrow Y) = \frac{s(X \cup Y)}{s(X)} = 0,8$$

¿Qué pasaría si el soporte a priori del itemset Y ya era 80%?

¿Y si hubiese sido mayor que 80%? ¿Y si hubiese sido menor que 80%?

Confianza

$$S(Y)=0,5$$

$$S(X)=0,8$$

$$S(X \cup Y)=0,4$$

$$C(X \Rightarrow Y)=0,5$$

$$C(Y \Rightarrow X)=0,8$$



Confianza

$$S(Y)=0,5$$

$$S(X)=0,6$$

$$S(X \cup Y)=0,4$$

$$C(X \Rightarrow Y)=0,6$$

$$C(Y \Rightarrow X)=0,8$$



Confianza

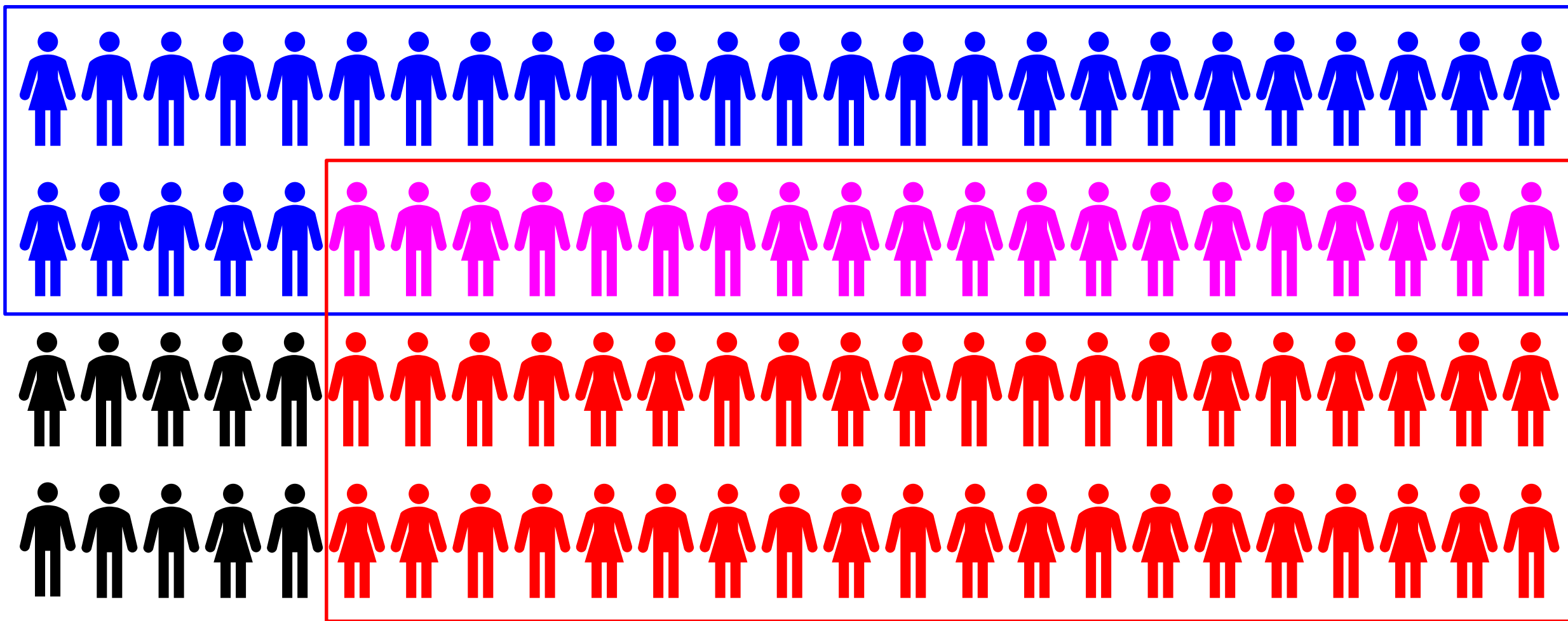
$$S(Y)=0,5$$

$$S(X)=0,6$$

$$S(X \cup Y)=0,2$$

$$C(X \Rightarrow Y)=0,3$$

$$C(Y \Rightarrow X)=0,4$$



Lift

Permite medir el cambio del lado derecho de la regla (consecuente Y) dada la presencia del lado izquierdo de la regla (antecedente X)

$$L(X \Rightarrow Y) = \frac{C(X \Rightarrow Y)}{S(Y)}$$

La magnitud del lift entrega indicios del efecto de la ocurrencia del antecedente en la ocurrencia del consecuente

Ejemplo de Lift

Transacción	Itemset
1	{ A , B }
2	{ A , C , D }
3	{ B , C , D , E }
4	{ A , B , C , D }
5	{ A , B , C , E }

$$S(\{ B \}) = 0,8$$

$$C(\{ A , D \} \Rightarrow \{ B \}) = 0,5$$

$$S(\{ C \}) = 0,8$$

$$C(\{ A , D \} \Rightarrow \{ C \}) = 1$$

$$S(\{ E \}) = 0,4$$

$$C(\{ A , D \} \Rightarrow \{ E \}) = 0$$

$$L(\{ A , D \} \Rightarrow \{ B \}) = 0,625$$

$$L(\{ A , D \} \Rightarrow \{ C \}) = 1,25$$

$$L(\{ A , D \} \Rightarrow \{ E \}) = 0$$

Lift

$$L > 1$$

La probabilidad del
consecuente de la regla
aumenta dado que el
consumidor compró los
ítems del antecedente

$$L = 1$$

La probabilidad del
consecuente de la regla
no se ve afectada por el
antecedente (i.e. son
independientes)

$$L < 1$$

El antecedente tuvo un
efecto negativo en la
ocurrencia del consecuente,
reduciendo su probabilidad

Algoritmo Apriori

Algoritmo Apriori

Este algoritmo permite encontrar reglas de asociación que cumplan con criterios de soporte y confianza de manera automática

El algoritmo funciona en forma iterativa, aumentando en cada iteración el tamaño de los itemsets candidatos a generar reglas de asociación, según su soporte

El algoritmo continua hasta que no se pueden generar nuevos itemsets

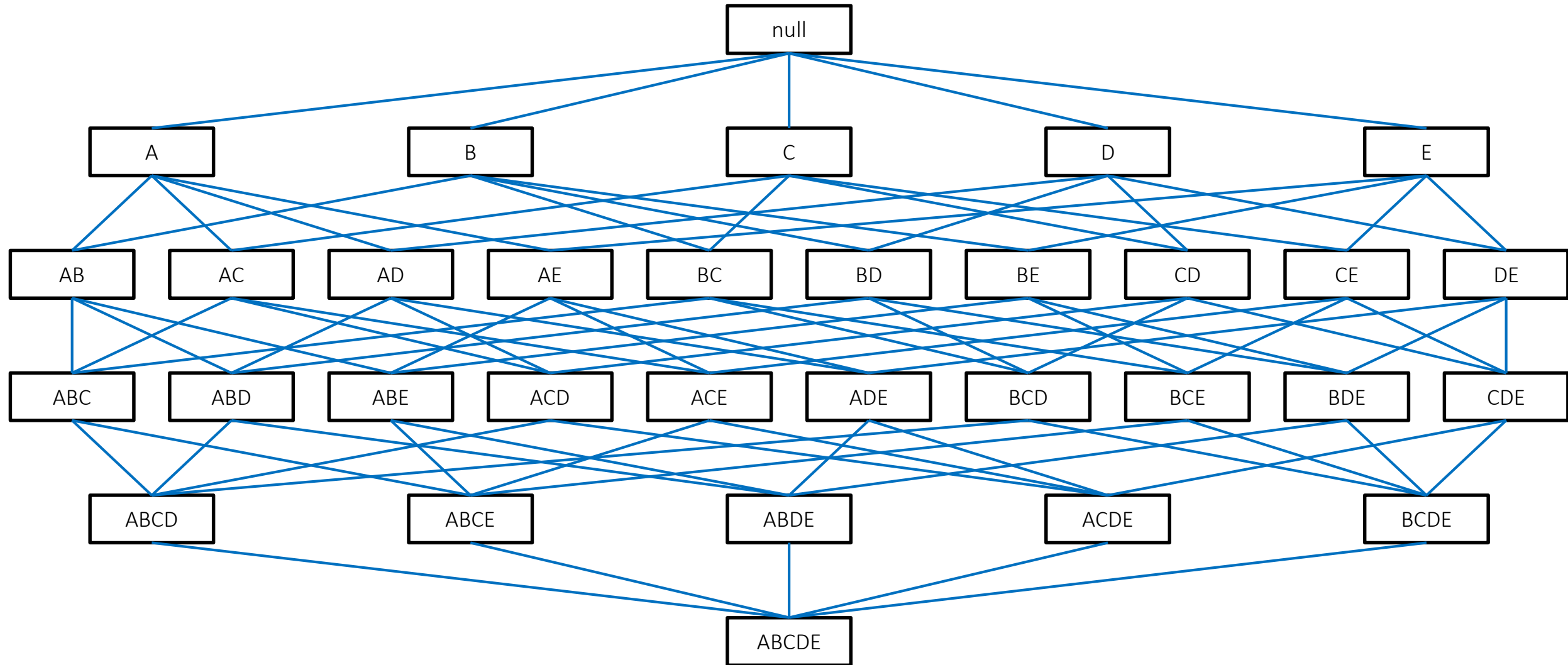
Generando itemsets

Una idea es obtener todos los posibles itemsets que se pueden formar a partir de un conjunto de ítems

Ejemplo: sea el conjunto $I = \{ A , B , C , D , E \}$, ¿cuántos itemsets se pueden generar?

Existen $2^5 - 1 = 31$ itemsets no vacíos

Generando itemsets



Generando itemsets

¿Qué pasa cuando la cantidad n de ítems es muy grande?

2^n se vuelve muy (muy pero muy muy) grande

Con tan solo 100 ítems la cantidad de itemsets es mayor a un quintillón

$$2^{100} = 1.267.650.600.228.229.401.496.703.205.376$$

Principio de monotonicidad

Podemos reducir la cantidad de itemsets a considerar haciendo uso del principio de monotonicidad

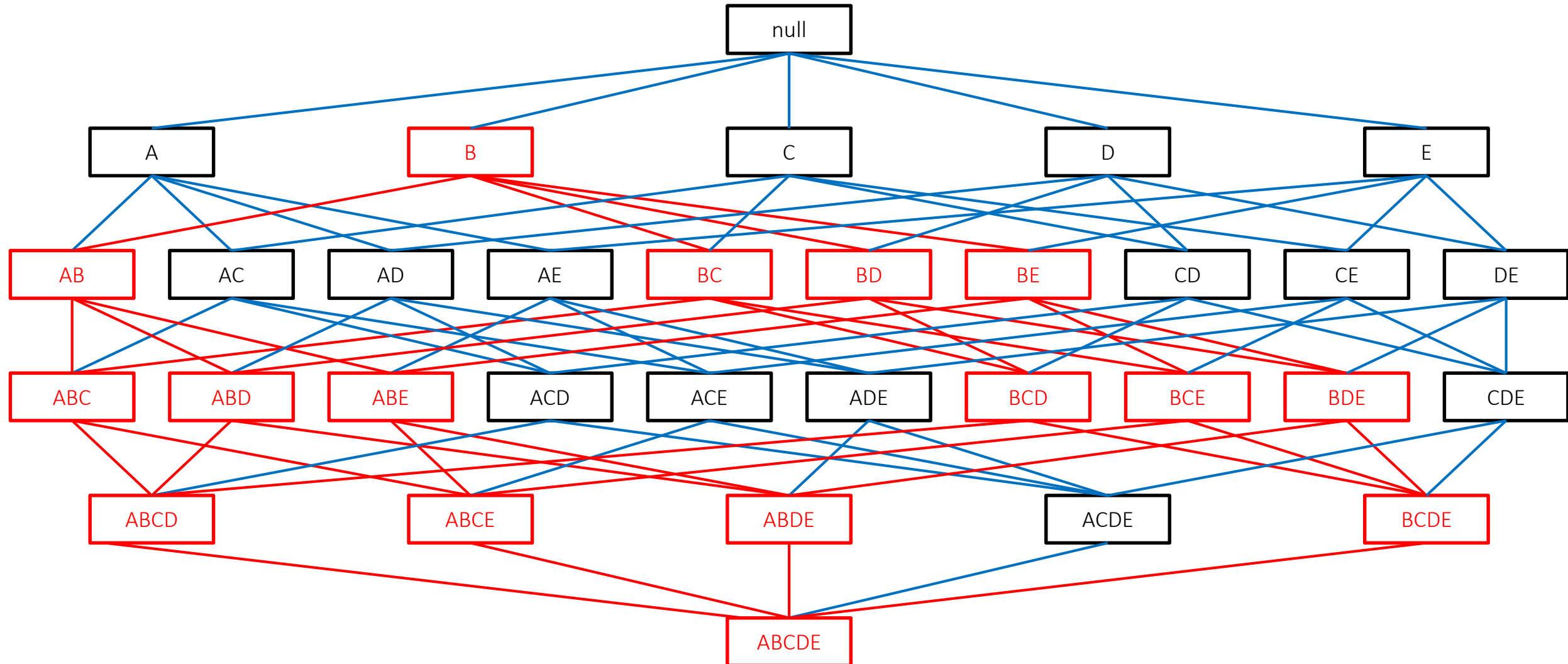
Sea $Y \subseteq X$, sabemos que $S(X) \leq S(Y)$

Si X es frecuente, entonces Y es frecuente

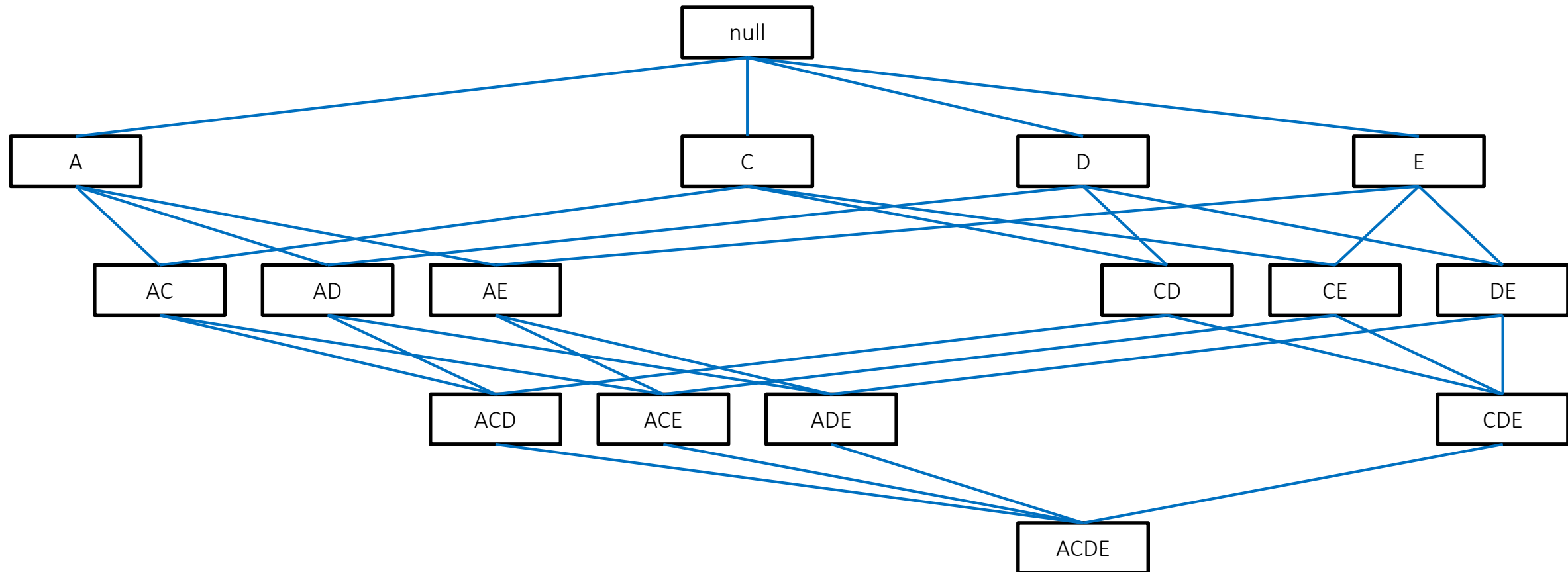
Si Y no es frecuente, entonces X no es frecuente

Principio de monotonicidad

{ B } infrecuente

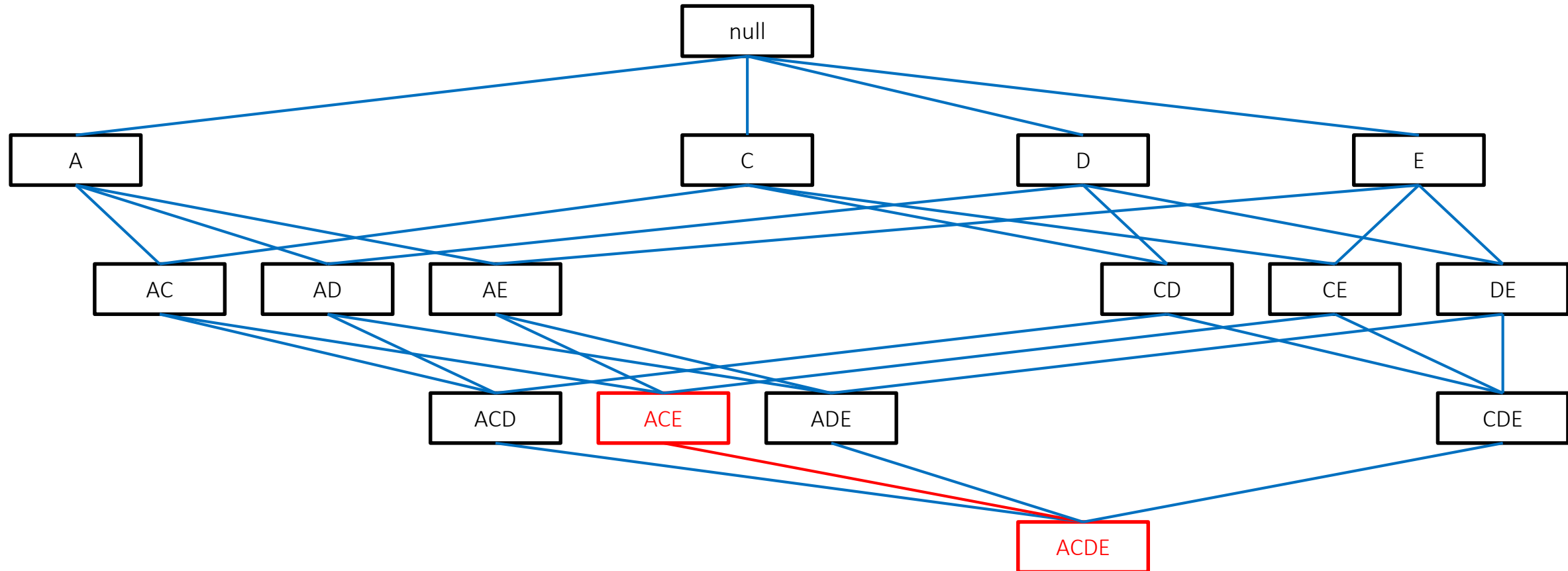


Principio de monotonicidad

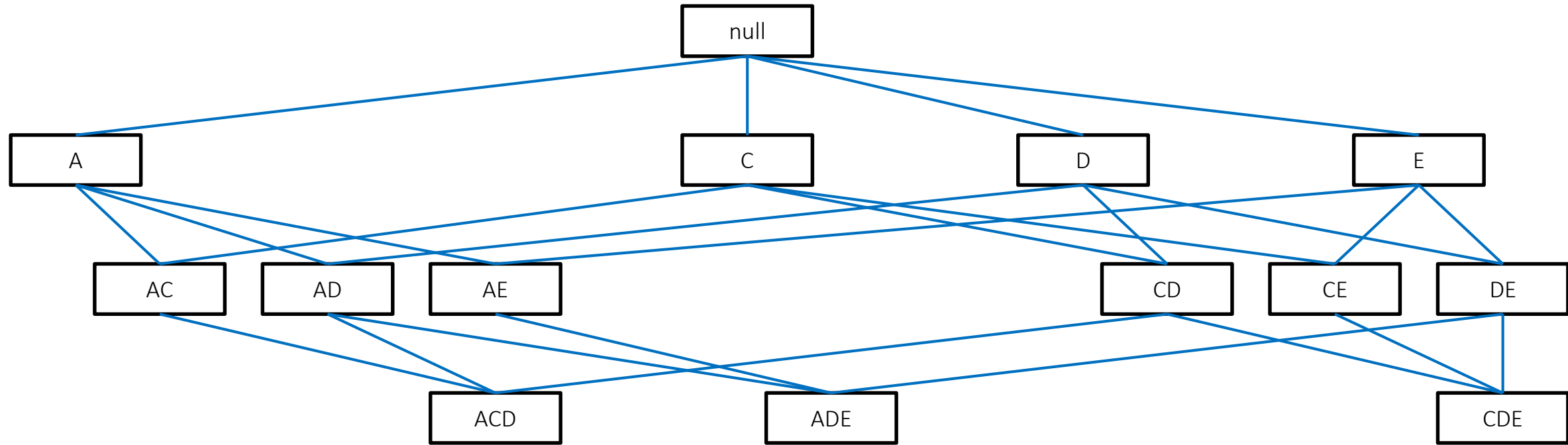


Principio de monotonicidad

{ A , C , E } infrecuente



Principio de monotonicidad



Consideremos los siguientes datos

Transacción	Itemset
1	{ A , B , F }
2	{ B , D }
3	{ B , C }
4	{ A , B , D }
5	{ A , C , F }
6	{ B , C , E }
7	{ A , C }
8	{ A , B , D , F }
9	{ A , B , C }

Obtendremos reglas de asociación
con un soporte mínimo de 2/9

Itemsets de tamaño 1

Transacción	Itemset
1	{ A , B , F }
2	{ B , D }
3	{ B , C }
4	{ A , B , D }
5	{ A , C , F }
6	{ B , C , E }
7	{ A , C }
8	{ A , B , D , F }
9	{ A , B , C }

Itemset	Soporte
{ A }	6/9
{ B }	7/9
{ C }	5/9
{ D }	3/9
{ E }	1/9 
{ F }	3/9

Itemsets de tamaño 1

$$L_1 = [\{A\}, \{B\}, \{C\}, \{D\}, \{F\}]$$

Generamos candidatos de tamaño 2

$$C_2 = [\{A, B\}, \{A, C\}, \{A, D\}, \{A, F\}, \{B, C\}, \\ \{B, D\}, \{B, F\}, \{C, D\}, \{C, F\}, \{D, F\}]$$

Itemsets de tamaño 2

Transacción	Itemset
1	{ A , B , F }
2	{ B , D }
3	{ B , C }
4	{ A , B , D }
5	{ A , C , F }
6	{ B , C , E }
7	{ A , C }
8	{ A , B , D , F }
9	{ A , B , C }

Itemset	Soporte	
{ A , B }	4/9	
{ A , C }	3/9	
{ A , D }	2/9	
{ A , F }	3/9	
{ B , C }	3/9	
{ B , D }	3/9	
{ B , F }	2/9	
{ C , D }	0	✗
{ C , F }	1/9	✗
{ D , F }	1/9	✗

Itemsets de tamaño 2

$$L_2 = [\{A, B\}, \{A, C\}, \{A, D\}, \{A, F\}, \{B, C\}, \{B, D\}, \{B, F\}]$$

Generamos candidatos de tamaño 3

$$C_3 = [\{A, B, C\}, \{A, B, D\}, \{A, B, F\}, \{A, C, D\}, \{A, C, F\}, \\ \{A, D, F\}, \{B, C, D\}, \{B, C, F\}, \{B, D, F\}]$$

Itemsets de tamaño 3

Transacción	Itemset
1	{ A , B , F }
2	{ B , D }
3	{ B , C }
4	{ A , B , D }
5	{ A , C , F }
6	{ B , C , E }
7	{ A , C }
8	{ A , B , D , F }
9	{ A , B , C }

Itemset	Soporte	
{ A , B , C }	1/9	✗
{ A , B , D }	2/9	
{ A , B , F }	2/9	
{ A , C , D }	0	✗
{ A , C , F }	1/9	✗
{ A , D , F }	1/9	✗
{ B , C , D }	0	✗
{ B , C , F }	0	✗
{ B , D , F }	1/9	✗

Itemsets de tamaño 3


$$L_3 = [\{A, B, D\}, \{A, B, F\}]$$

Generamos candidatos de tamaño 4

$$C_4 = [\{A, B, D, F\}]$$

Itemsets de tamaño 4

Transacción	Itemset
1	{ A , B , F }
2	{ B , D }
3	{ B , C }
4	{ A , B , D }
5	{ A , C , F }
6	{ B , C , E }
7	{ A , C }
8	{ A , B , D , F }
9	{ A , B , C }

Itemset	Soporte
{ A , B , D , F }	1/9 

No hay itemsets de tamaño 4 que cumplan el criterio de soporte

Terminamos de obtener itemsets

Itemsets más frecuentes

Los siguientes itemsets cumplen el criterio de soporte establecido

$$L_1 = [\{A\}, \{B\}, \{C\}, \{D\}, \{F\}]$$

$$L_2 = [\{A, B\}, \{A, C\}, \{A, D\}, \{A, F\}, \{B, C\}, \{B, D\}, \{B, F\}]$$

$$L_3 = [\{A, B, D\}, \{A, B, F\}]$$

Reglas de asociación resultantes

A partir de los itemsets más frecuentes podemos generar reglas de asociación

Tomemos por ejemplo los itemsets de tamaño 3

$$L_3 = [\{ A , B , D \} , \{ A , B , F \}]$$

A partir de estos dos itemsets podemos obtener seis reglas de asociación

Reglas de asociación resultantes

Itemset	Regla	Soporte Itemset	Soporte Componentes	Confianza Regla	Lift Regla
{ A , B , D }	{ A , B } \Rightarrow { D }	2/9	4/9 \Rightarrow 3/9	1/2	3/2
{ A , B , D }	{ A , D } \Rightarrow { B }	2/9	2/9 \Rightarrow 7/9	1	9/7
{ A , B , D }	{ B , D } \Rightarrow { A }	2/9	3/9 \Rightarrow 6/9	2/3	1
{ A , B , F }	{ A , B } \Rightarrow { F }	2/9	4/9 \Rightarrow 3/9	1/2	3/2
{ A , B , F }	{ A , F } \Rightarrow { B }	2/9	3/9 \Rightarrow 7/9	2/3	6/7
{ A , B , F }	{ B , F } \Rightarrow { A }	2/9	2/9 \Rightarrow 6/9	1	3/2

Podemos utilizar la confianza para seleccionar las “mejores” reglas

Reglas de asociación resultantes

Todas las reglas de asociación con soporte $\geq 2/9$ y confianza $\geq 2/3$ son:

Itemset	Regla	Soporte Itemset	Soporte Componentes	Confianza Regla	Lift Regla
{ A }	{ } \Rightarrow { A }	6/9	1 \Rightarrow 6/9	6/9	1
{ B }	{ } \Rightarrow { B }	7/9	1 \Rightarrow 7/9	7/9	1
{ A , B }	{ A } \Rightarrow { B }	4/9	6/9 \Rightarrow 7/9	2/3	6/7
{ A , D }	{ D } \Rightarrow { A }	2/9	3/9 \Rightarrow 6/9	2/3	1
{ A , F }	{ F } \Rightarrow { A }	3/9	3/9 \Rightarrow 6/9	1	3/2
{ B , D }	{ D } \Rightarrow { B }	3/9	3/9 \Rightarrow 7/9	1	9/7
{ B , F }	{ F } \Rightarrow { B }	2/9	3/9 \Rightarrow 7/9	2/3	6/7
{ A , B , D }	{ A , D } \Rightarrow { B }	2/9	2/9 \Rightarrow 7/9	1	9/7
{ A , B , D }	{ B , D } \Rightarrow { A }	2/9	3/9 \Rightarrow 6/9	2/3	1
{ A , B , F }	{ A , F } \Rightarrow { B }	2/9	3/9 \Rightarrow 7/9	2/3	6/7
{ A , B , F }	{ B , F } \Rightarrow { A }	2/9	2/9 \Rightarrow 6/9	1	3/2

Algunas limitaciones de Apriori

La generación de candidatos puede ser extremadamente lenta (pares, tripletes, etc.)

El algoritmo recorre todas las transacciones en cada iteración

Ítems en formato de strings hacen al algoritmo mucho más pesado

Alto consumo de memoria

Algoritmo FP-Growth

Algoritmo FP-Growth

FP = *Frecuent Pattern*

Busca mejorar algunas limitaciones del algoritmo A priori

Se basa en una estructura de árbol llamada FP-Tree

Cada nodo representa un ítem con su cuenta actual

Cada rama representa una asociación diferente

Consideremos los siguientes datos

Transacción	Itemset
1	{ A , B , F }
2	{ B , D }
3	{ B , C }
4	{ A , B , D }
5	{ A , C , F }
6	{ B , C , E }
7	{ A , C }
8	{ A , B , D , F }
9	{ A , B , C }

Obtendremos reglas de asociación
con un soporte mínimo de 2/9

Obtenemos el soporte de cada ítem

Transacción	Itemset
1	{ A , B , F }
2	{ B , D }
3	{ B , C }
4	{ A , B , D }
5	{ A , C , F }
6	{ B , C , E }
7	{ A , C }
8	{ A , B , D , F }
9	{ A , B , C }

Itemset	Soporte
{ A }	6/9
{ B }	7/9
{ C }	5/9
{ D }	3/9
{ E }	1/9 
{ F }	3/9

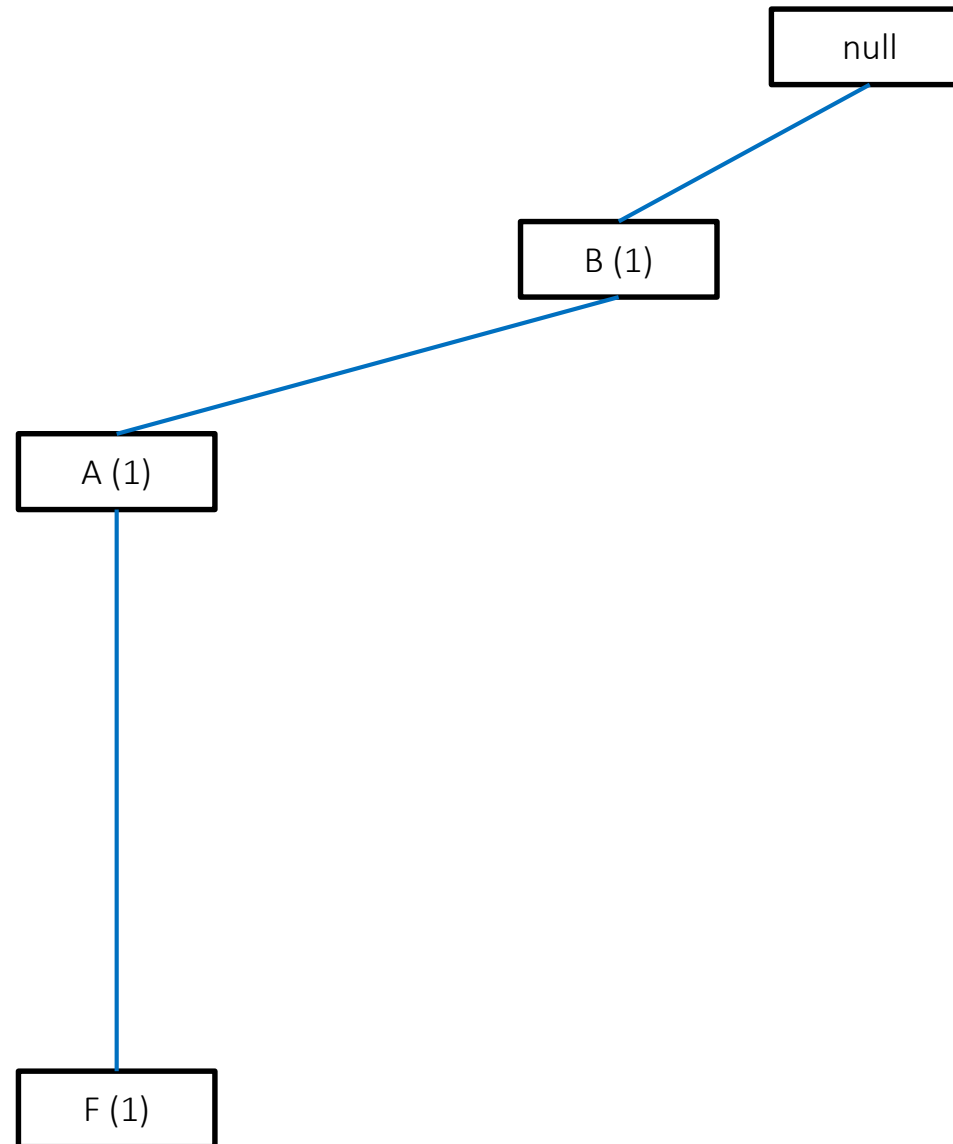
Ordenamos las transacciones según el soporte de sus ítems

Itemset	Soporte
{ B }	7/9
{ A }	6/9
{ C }	5/9
{ D }	3/9
{ F }	3/9

Transacción	Itemset	Itemset Ordenado
1	{ A , B , F }	{ B , A , F }
2	{ B , D }	{ B , D }
3	{ B , C }	{ B , C }
4	{ A , B , D }	{ B , A , D }
5	{ A , C , F }	{ A , C , F }
6	{ B , C , E }	{ B , C }
7	{ A , C }	{ A , C }
8	{ A , B , D , F }	{ B , A , D , F }
9	{ A , B , C }	{ B , A , C }

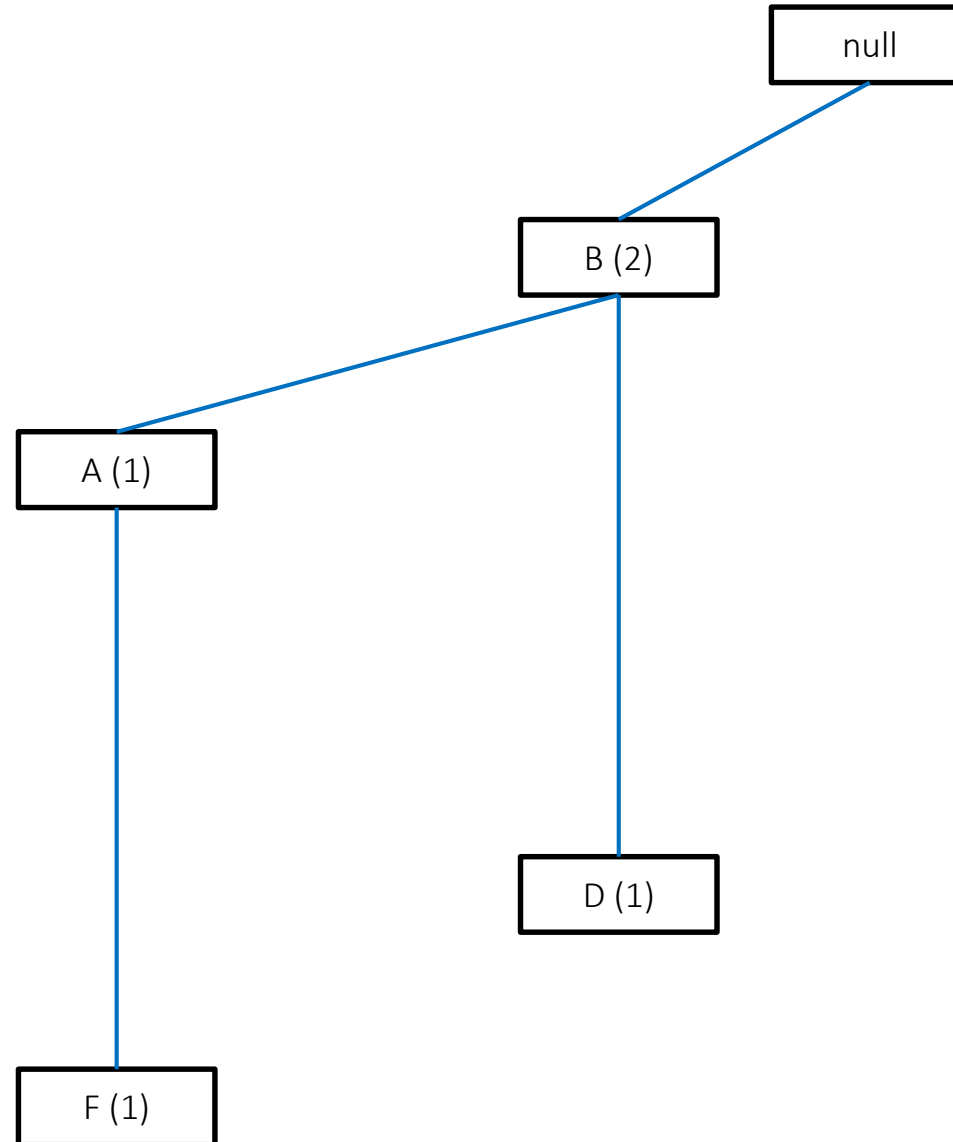
Construimos el árbol

<i>T</i>	Itemset Ordenado
1	{ B , A , F }
2	{ B , D }
3	{ B , C }
4	{ B , A , D }
5	{ A , C , F }
6	{ B , C }
7	{ A , C }
8	{ B , A , D , F }
9	{ B , A , C }



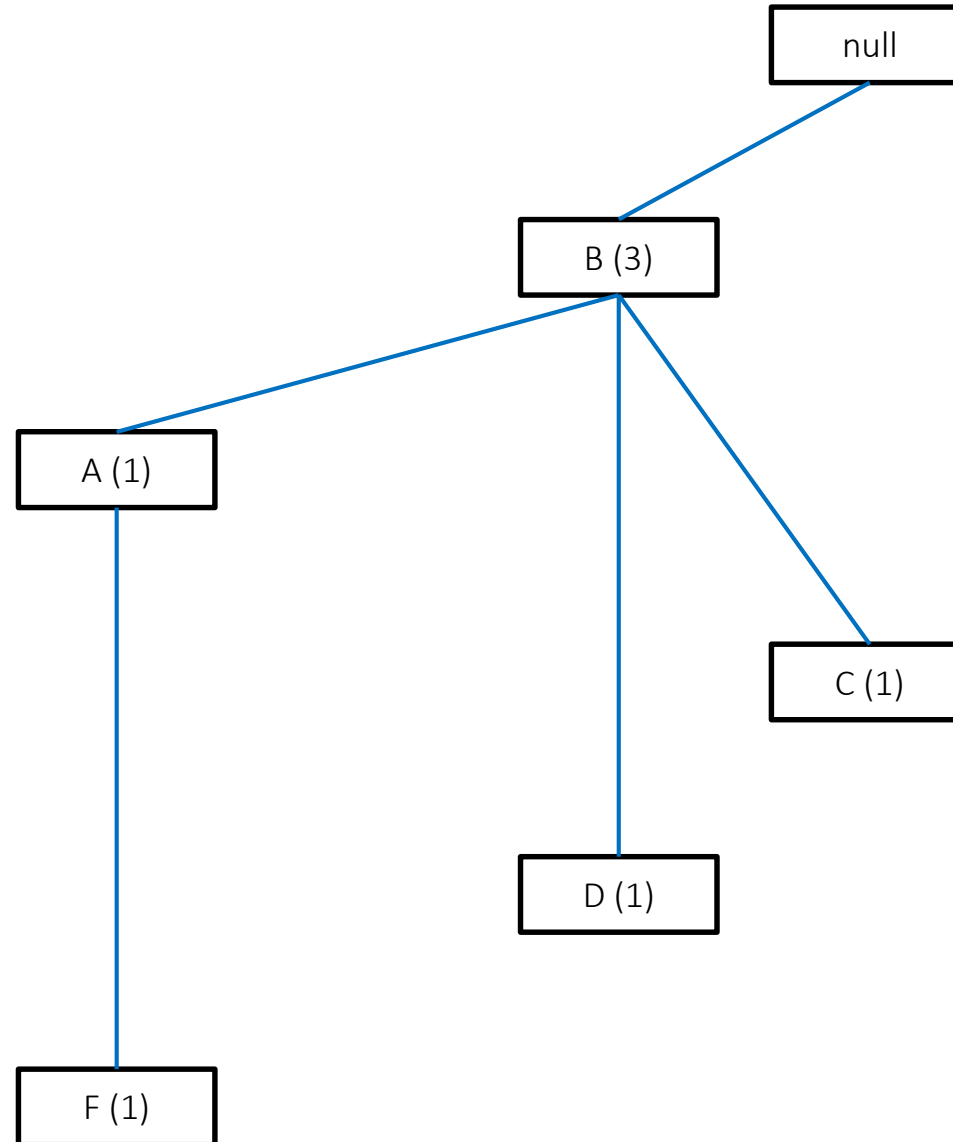
Construimos el árbol

<i>T</i>	Itemset Ordenado
1	{ B , A , F }
2	{ B , D }
3	{ B , C }
4	{ B , A , D }
5	{ A , C , F }
6	{ B , C }
7	{ A , C }
8	{ B , A , D , F }
9	{ B , A , C }



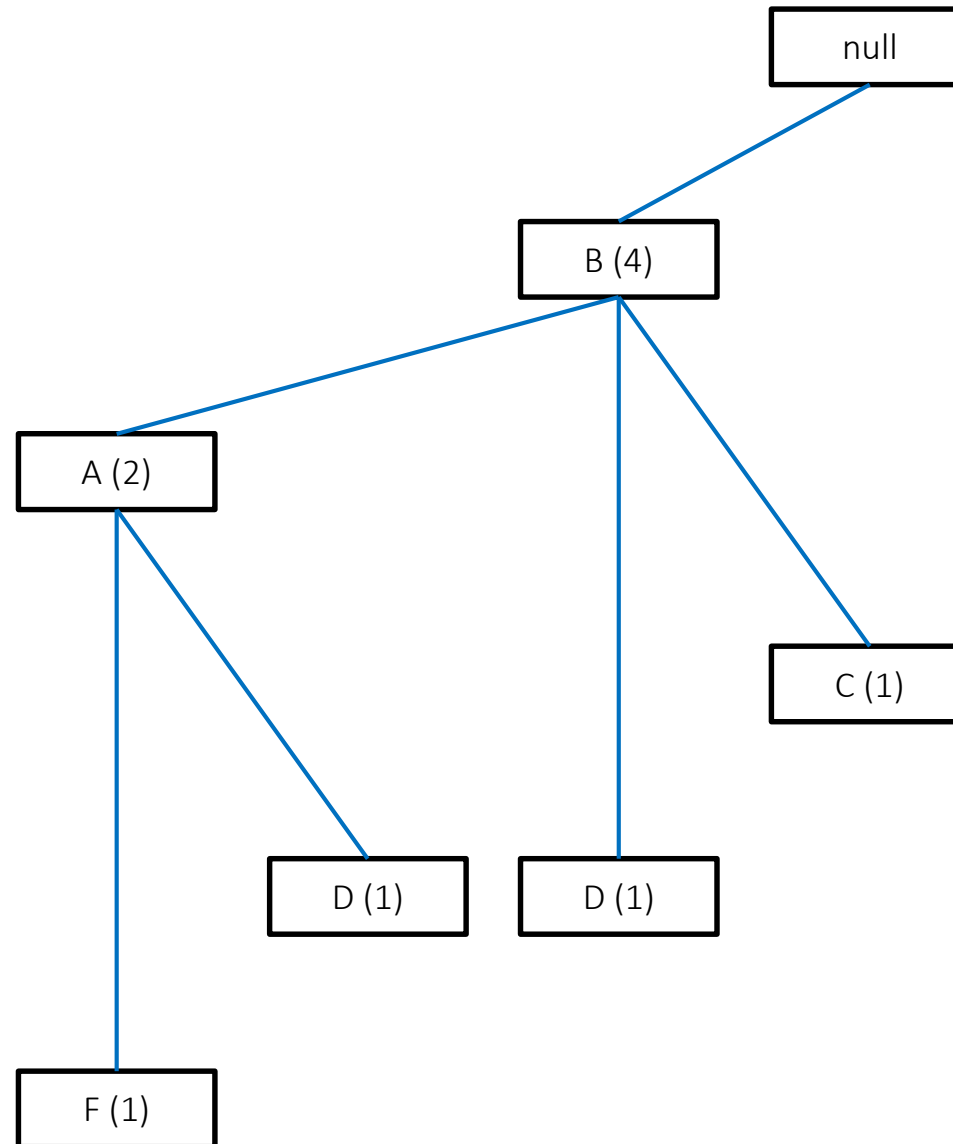
Construimos el árbol

<i>T</i>	Itemset Ordenado
1	{ B , A , F }
2	{ B , D }
3	{ B , C }
4	{ B , A , D }
5	{ A , C , F }
6	{ B , C }
7	{ A , C }
8	{ B , A , D , F }
9	{ B , A , C }



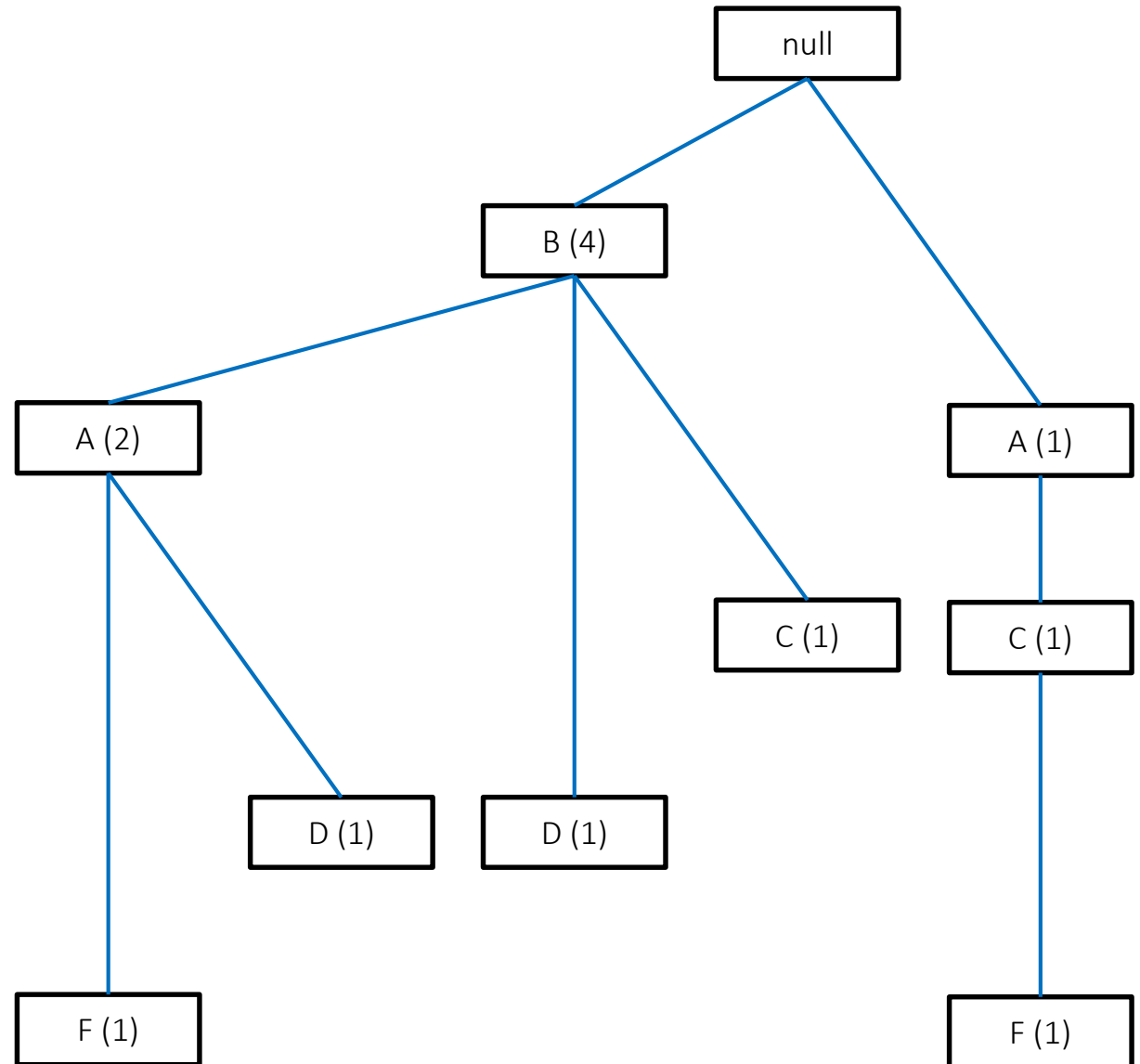
Construimos el árbol

<i>T</i>	Itemset Ordenado
1	{ B , A , F }
2	{ B , D }
3	{ B , C }
4	{ B , A , D }
5	{ A , C , F }
6	{ B , C }
7	{ A , C }
8	{ B , A , D , F }
9	{ B , A , C }



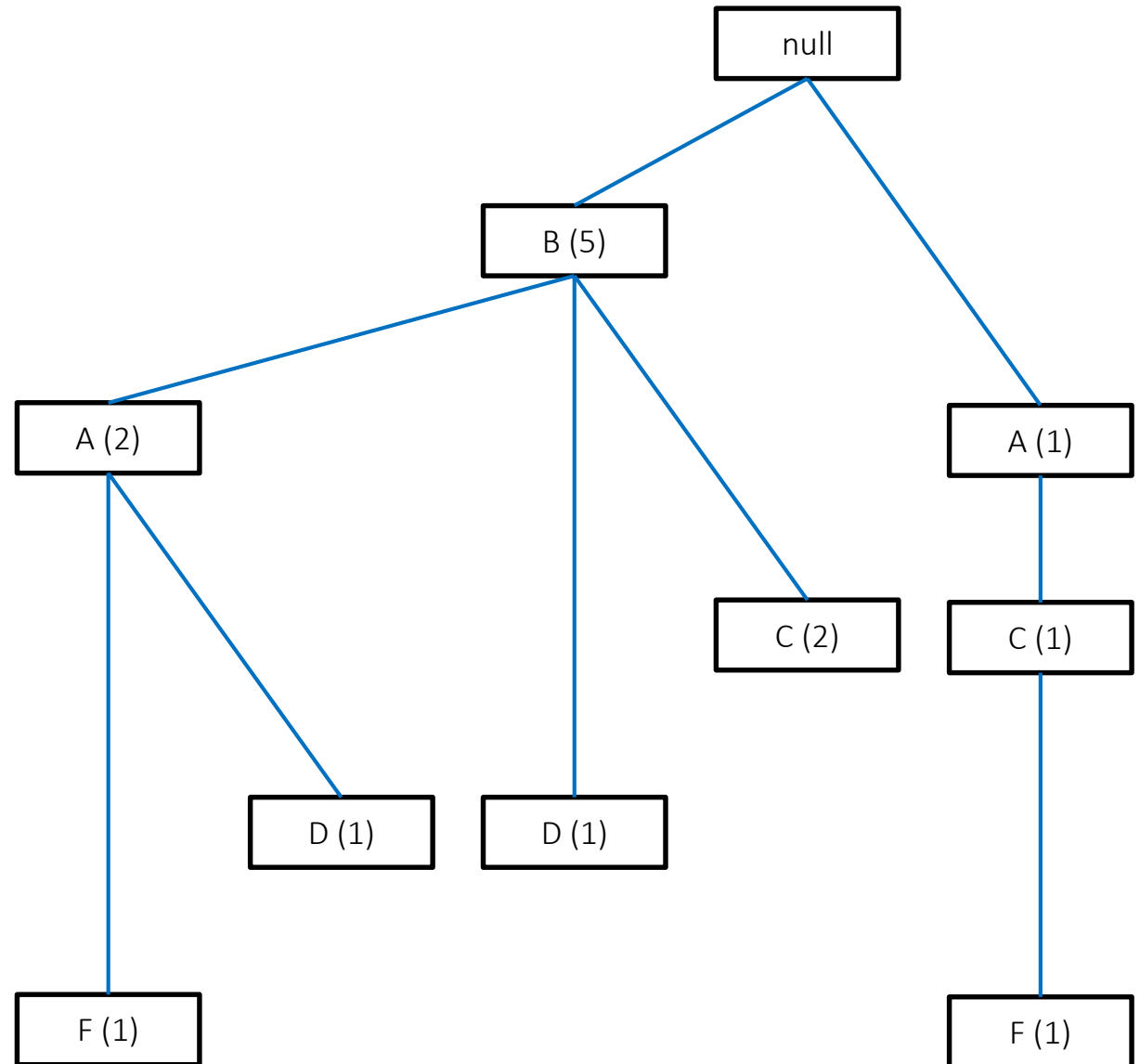
Construimos el árbol

<i>T</i>	Itemset Ordenado
1	{ B , A , F }
2	{ B , D }
3	{ B , C }
4	{ B , A , D }
5	{ A , C , F }
6	{ B , C }
7	{ A , C }
8	{ B , A , D , F }
9	{ B , A , C }



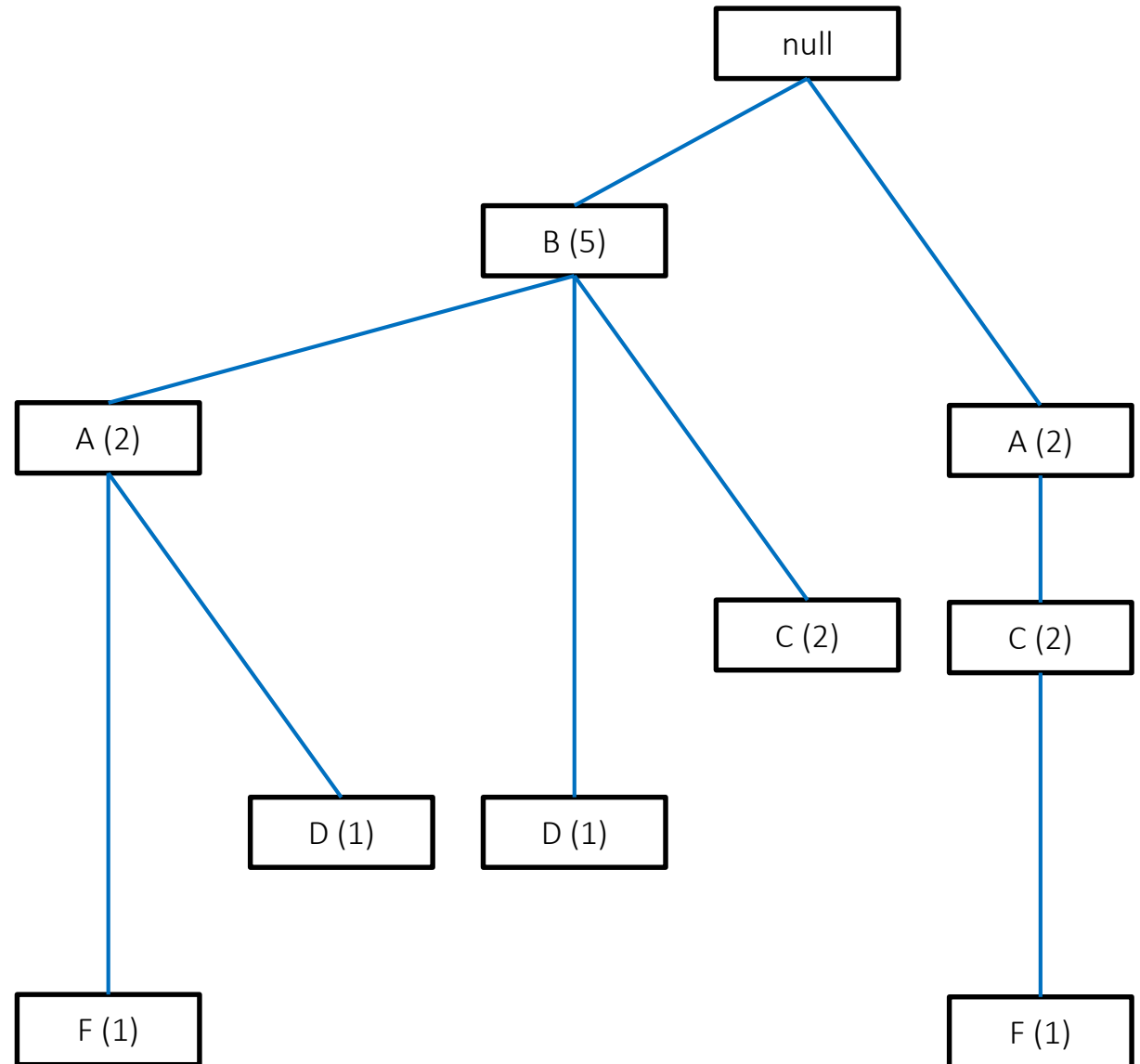
Construimos el árbol

<i>T</i>	Itemset Ordenado
1	{ B , A , F }
2	{ B , D }
3	{ B , C }
4	{ B , A , D }
5	{ A , C , F }
6	{ B , C }
7	{ A , C }
8	{ B , A , D , F }
9	{ B , A , C }



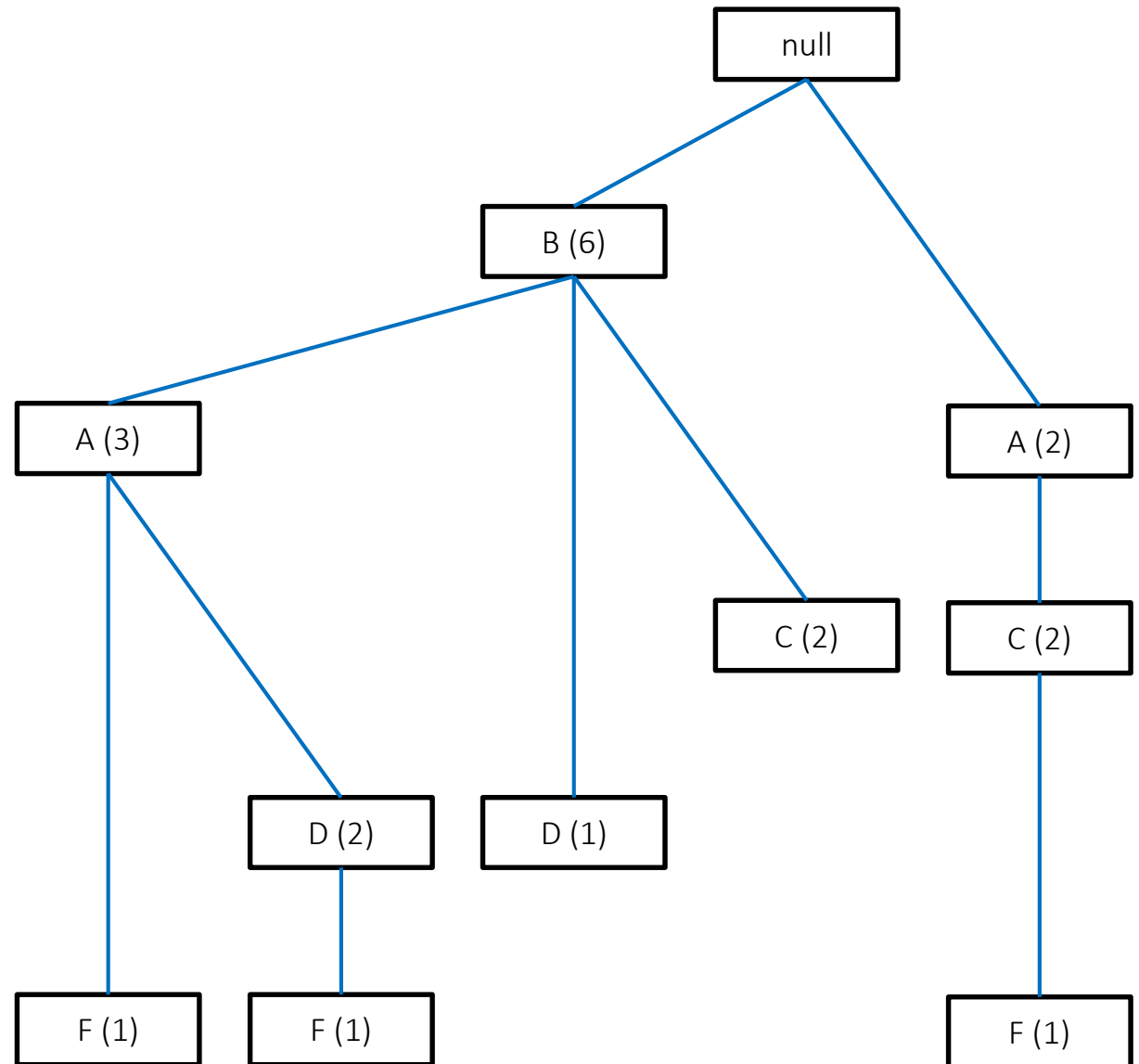
Construimos el árbol

<i>T</i>	Itemset Ordenado
1	{ B , A , F }
2	{ B , D }
3	{ B , C }
4	{ B , A , D }
5	{ A , C , F }
6	{ B , C }
7	{ A , C }
8	{ B , A , D , F }
9	{ B , A , C }



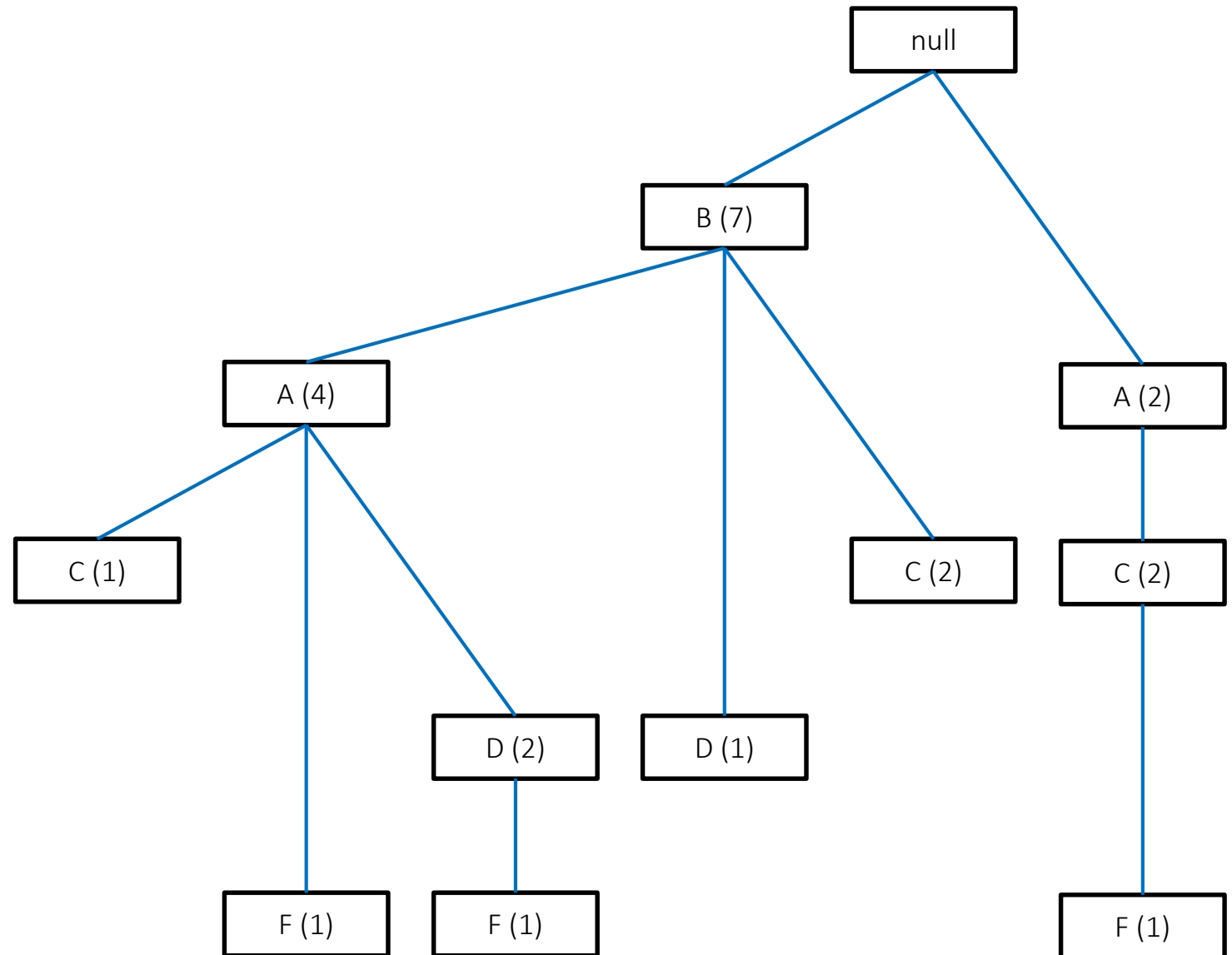
Construimos el árbol

<i>T</i>	Itemset Ordenado
1	{ B , A , F }
2	{ B , D }
3	{ B , C }
4	{ B , A , D }
5	{ A , C , F }
6	{ B , C }
7	{ A , C }
8	{ B , A , D , F }
9	{ B , A , C }



Construimos el árbol

<i>T</i>	Itemset Ordenado
1	{ B , A , F }
2	{ B , D }
3	{ B , C }
4	{ B , A , D }
5	{ A , C , F }
6	{ B , C }
7	{ A , C }
8	{ B , A , D , F }
9	{ B , A , C }



Patrones condicionales

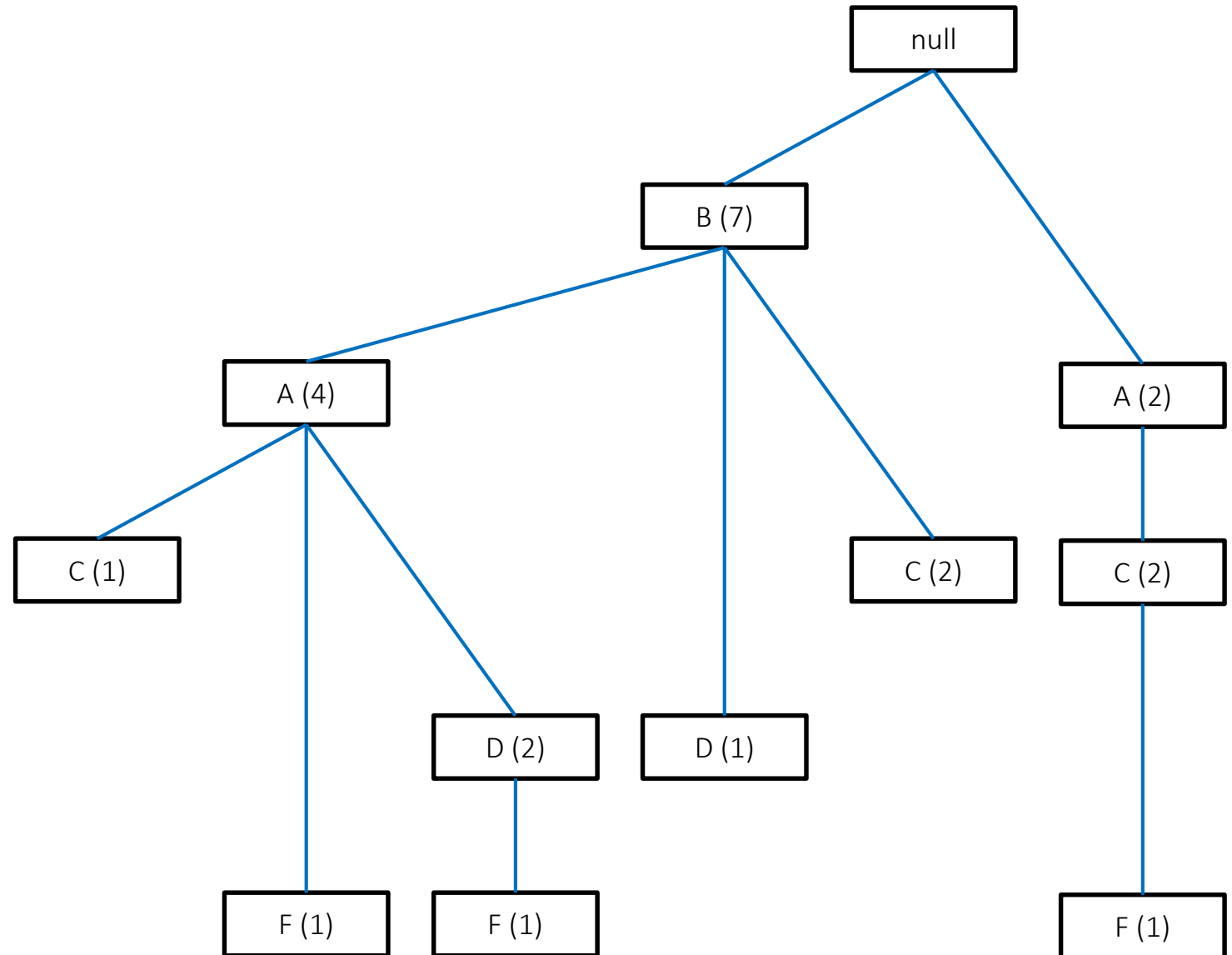
En orden reverso de soporte

Ítem F

{ B , A } (1)

{ B , A , D } (1)

{ A , C } (1)



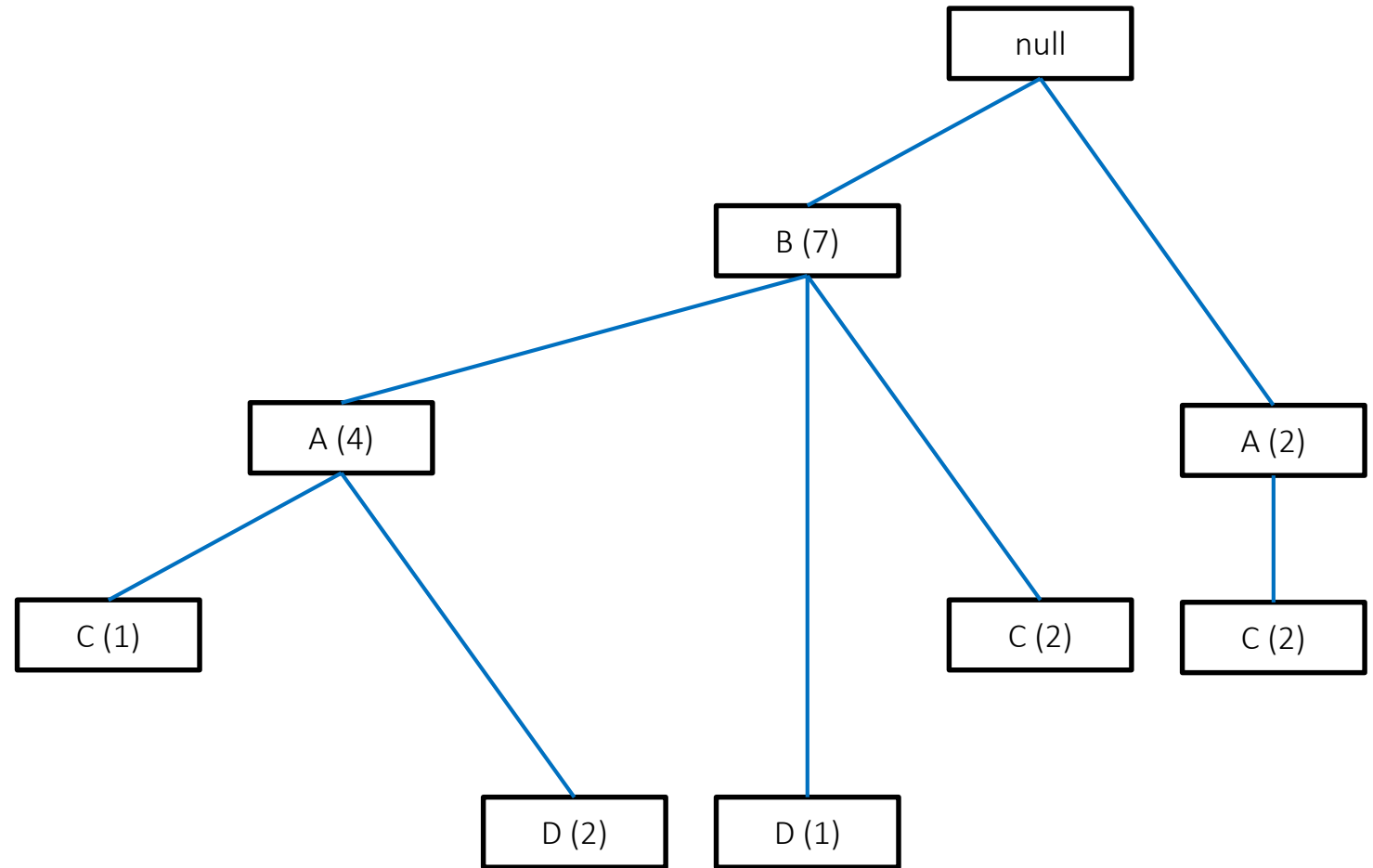
Patrones condicionales

En orden reverso de soporte

Ítem D

{ B , A } (2)

{ B } (1)



Patrones condicionales

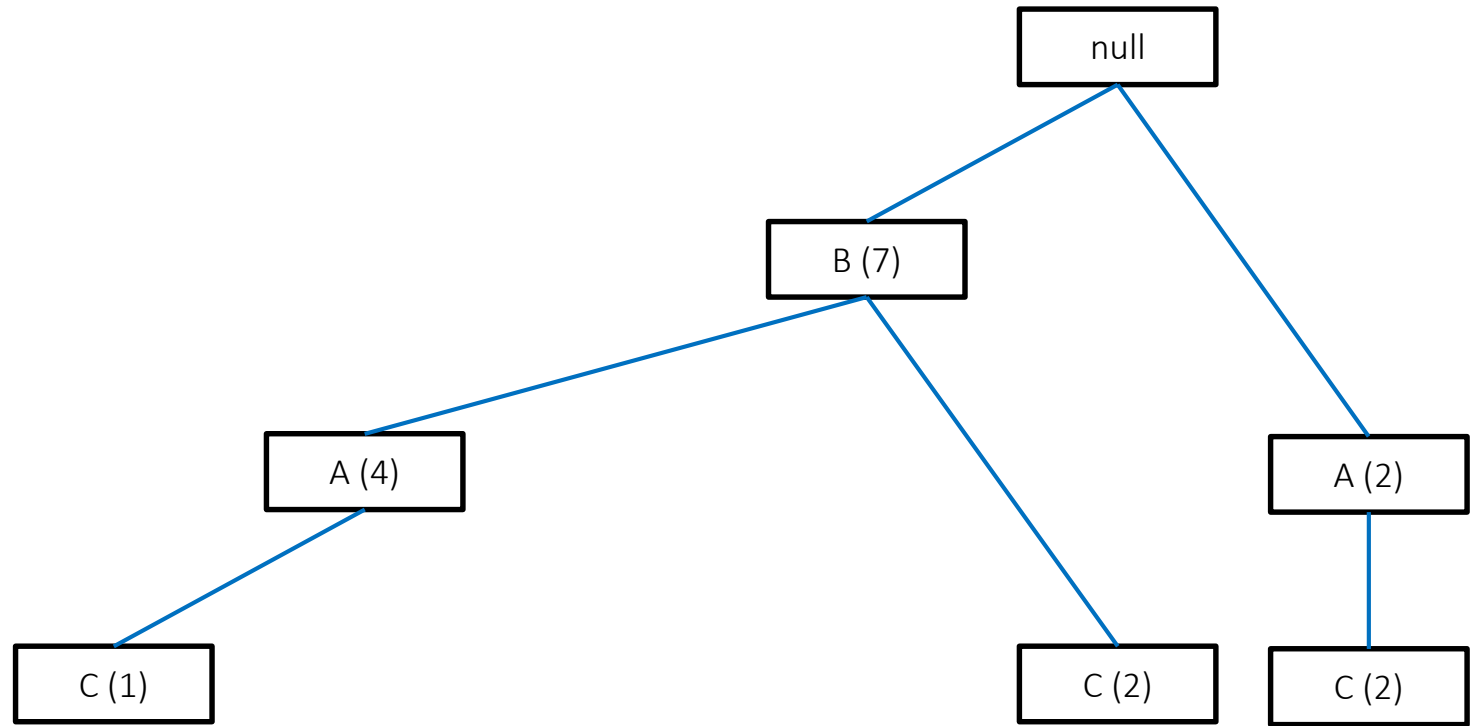
En orden reverso de soporte

Ítem C

{ B , A } (1)

{ B } (2)

{ A } (2)



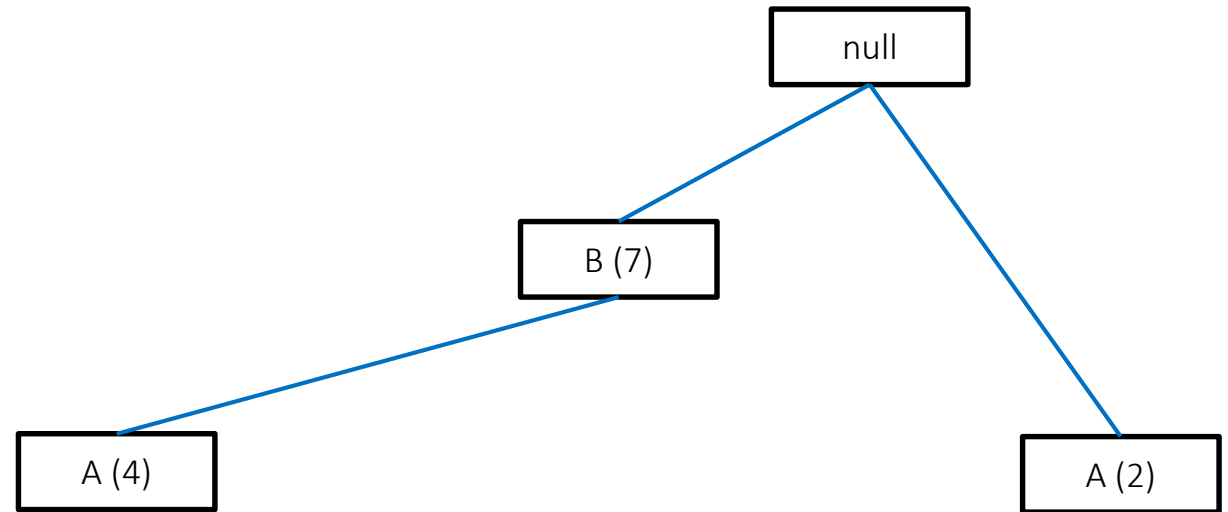
Patrones condicionales

En orden reverso de soporte

Ítem A

{ B } (4)

{ - } (2)

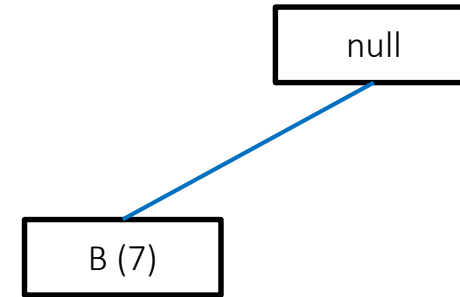


Patrones condicionales

En orden reverso de soporte

Ítem B

{ - } (7)



Patrones condicionales

Ítem	Patrones Condicionales	FP-Tree Condicional
F	$\{B, A\}(1), \{B, A, D\}(1), \{A, C\}(1)$	$\{B, A\}(2), \{B\}(2), \{A\}(3)$
D	$\{B, A\}(2), \{B\}(1)$	$\{B, A\}(2), \{B\}(3), \{A\}(2)$
C	$\{B, A\}(1), \{B\}(2), \{A\}(2)$	$\{B\}(3), \{A\}(3)$
A	$\{B\}(4)$	$\{B\}(4)$
B	-	-

Recordar que estamos trabajando con un soporte mínimo de 2/9
(i.e. una ocurrencia mínima de 2)

Patrones condicionales

Ítem	FP-Tree Condicional	Itemsets Frecuentes
F	$\{B, A\}, \{B\}, \{A\}$	$\{A, B, F\}, \{B, F\}, \{A, F\}$
D	$\{B, A\}, \{B\}, \{A\}$	$\{A, B, D\}, \{B, D\}, \{A, D\}$
C	$\{B\}, \{A\}$	$\{B, C\}, \{A, C\}$
A	$\{B\}$	$\{A, B\}$
B	-	-

Obtenemos los mismos itemsets de tamaño 2 y 3 que habíamos obtenido con Apriori

Ahora podríamos generar reglas de asociación