

# Reglas de Asociación

Tratamiento Inteligente de Datos  
Master Universitario en Ingeniería Informática



UNIVERSIDAD  
DE GRANADA

Gabriel Navarro ([gnavarro@ugr.es](mailto:gnavarro@ugr.es), [gnavarro@decsai.ugr.es](mailto:gnavarro@decsai.ugr.es))

# Objetivos

- ❑ Entender el concepto de regla de asociación y su utilidad como tarea descriptiva de Minería de Datos
- ❑ Conocer el algoritmo Apriori como generador de itemsets frecuentes
- ❑ Conocer el algoritmo FP-Growth como generador de itemsets frecuentes
- ❑ Conocer algunas medidas de evaluación de reglas
- ❑ Conocer el concepto de regla de asociación difusa

# Índice

- Concepto de regla de asociación
- Algoritmo Apriori
- Variaciones del algoritmo A priori
- Algoritmo FP-Growth
- Generación de reglas
- Evaluación de reglas
- Reglas difusas
- Otros aspectos

# Problema básico



**APRENDIZAJE  
NO SUPERVISADO**

Encontrar reglas que relacionen la co-ocurrencia de ítems

SI plátanos **ENTONCES** zanahorias

# Reglas de asociación

## Objetivo primario

Descubrir "reglas" en lógica proposicional (pero cualificadas probabilísticamente) que involucren algunos valores de ciertos atributos.

- El formato de los datos es más general que una BD relacional.  
Se trabaja en una base de datos de transacciones

ID Cesta	Cesta
001	Lechuga, Embutido, Cerveza, Pan de Molde
010	Cerveza, Pañales, Queso, Lechuga, Champú
011	Pizza, Refresco, Palomitas
100	Pañales, Jabón, Champú

El 100% de los clientes que compran pañales, compran champú  
El 50% de los clientes que compran lechuga, compran embutido

# Reglas de asociación

La reglas de asociación son reglas del estilo

**SI** {item<sub>1</sub>,item<sub>2</sub>,...,item<sub>n</sub>} **ENTONCES** {item<sub>A</sub>,item<sub>B</sub>,..,item<sub>Z</sub>}

Por ejemplo,

ID Cesta	Cesta
001	Lechuga, Embutido, Cerveza, Pan de Molde
010	Cerveza, Pañales, Queso, Lechuga, Champú
011	Pizza, Refresco, Palomitas
100	Pañales, Jabón, Champú

**SI** Pañales **ENTONCES** Champú

**SI** Cerveza y Lechuga **ENTONCES** Embutido y Pan de Molde

# Reglas de asociación

SI Pañales ENTONCES Champú

- ❑ Las reglas no implican causalidad
  - No quiere decir que comprar Champú es consecuencia de comprar pañales
- ❑ Las reglas implican co-ocurrencia
  - Las personas compran Pañales y Champú a la vez

El sentido es distinto a las reglas de clasificación

# Reglas de asociación

Otro ejemplo:

- Una colección de documentos (BD trasaccional)
- Términos de cada documento (cesta de la compra)

Se obtiene

Si  $\{\text{amor, dolor}\}$  entonces  $\{\text{muerte}\}$

Las reglas no implican causalidad, si no co-ocurrencia

# Reglas de asociación

- ❑ Es una de las técnicas más utilizadas para expresar patrones de datos
- ❑ Normalmente, se busca un **conocimiento de los datos** para tomar decisiones
- ❑ Se suele encuadrar dentro de la **Minería de Conjuntos de Items Frecuentes** (Frequent Itemset Mining)
- ❑ Se pretende trabajar con gran cantidad de datos por lo que los algoritmos deben ser muy **eficientes**
- ❑ En su versión original, se trabaja con **atributos discretos**

# Modelo formal (Agrawal et al., 1993)

Sean  $I$  un conjunto de items (objetos) y  $T$  un conjunto de transacciones (subconjuntos de objetos de  $I$ ). Ambos son conjuntos finitos

Sean  $A$  y  $C$  subconjuntos de  $I$ , no vacíos y disjuntos. Diremos que existe en  $T$  una regla  $A \rightarrow C$  si y solo si cualquier transacción en  $T$  que contenga a  $A$  también contiene a  $C$

Obviamente, esto es demasiado estricto (100% de cumplimiento). Necesitamos medidas que nos indiquen la bondad de las asociaciones

# Definiciones

- **Itemset.** Conjunto de items (artículos)
- **K-itemset.** Itemset de cardinal k
- **Soporte** (support) de un itemset. Fraccion de las transacciones que contienen el itemset
- **Itemset frecuente.** Itemset con soporte igual o superior a un umbral de soporte establecido por el usuario (MinSupp)

# Definiciones

Por ejemplo,

ID Cesta	Cesta
001	Lechuga, Embutido, Cerveza, Pan de Molde
010	Cerveza, Pañales, Queso, Lechuga, Champú
011	Pizza, Refresco, Palomitas
100	Pañales, Jabón, Champú

- $\{\text{Lechuga}, \text{Cerveza}\}$  es un 2-itemset de soporte 0.5
- $\{\text{Jabón}\}$  es un 1-itemset de soporte 0.25
- $\{\text{Pañales}, \text{Lechuga}, \text{Embutido}\}$  es un 3-itemset de soporte 0

# Definiciones

Por tanto,

Una regla de asociación es un regla del tipo

**SI A ENTONCES B**

donde A y B son itemsets

También se suelen notar de la forma  $A \rightarrow B$

# Reglas de asociación

Podemos generalizar a atributos multivalorados para no restringir el problema solamente a la cesta de la compra

Outlook	Temp	Humidity	Windy
Rainy	Hot	High	False
Rainy	Hot	High	True
Overcast	Hot	High	False
Sunny	Mild	High	False
Sunny	Cool	Normal	False
Sunny	Cool	Normal	True
Overcast	Cool	Normal	True
Rainy	Mild	High	False
Rainy	Cool	Normal	False
Sunny	Mild	Normal	False
Rainy	Mild	Normal	True
Overcast	Mild	High	True
Overcast	Hot	Normal	False
Sunny	Mild	High	True

# Reglas de asociación

Outlook	Temp	Humidity	Windy
Rainy	Hot	High	False
Rainy	Hot	High	True
Overcast	Hot	High	False
Sunny	Mild	High	False
Sunny	Cool	Normal	False
Sunny	Cool	Normal	True
Overcast	Cool	Normal	True
Rainy	Mild	High	False
Rainy	Cool	Normal	False
Sunny	Mild	Normal	False
Rainy	Mild	Normal	True
Overcast	Mild	High	True
Overcast	Hot	Normal	False
Sunny	Mild	High	True

R: Temp=Hot → Humidity=High

$$\text{Supp}(R) = 3/14 = 0.21$$

$$\text{Conf}(R) = 3/4 = 0.75$$

# Reglas de asociación

En realidad, lo que se ha hecho es pasar un BD relacional a una BD transaccional

BDR:

DNI	Nombre	Altura	Peso	Dirección
5	JC	186	87	Gr
6	P	175	70	Ma

BDT:

```
5 NombreJC , Altura186 , Peso87 , DireccGr  
6 NombreP , Altura175 , Peso70 , DireccMa
```

# Reglas de asociación

Al revés,

BDT:

Cliente1	lechePascualE , azúcar1Kg , pepinos , cinta_8mm
Cliente2	lechePascualE , azúcar1Kg , ternera

↓ Recodificación

BDT:

Cliente1:	Lact3 , Ultr5 , Fruta9 , Otros15
Cliente2:	Lact3 , Ultr5 , Carne2

↓ Transformación

BDR:

IDCI	Lact	Ultr	Fruta	Carne	Pesc	Otros
Cliente1:	3	5	9	null	null	15
Cliente2:	3	5	null	2	null	null

los algoritmos que extraen reglas de asociación siempre se diseñan para que trabajen directamente sobre BDT

# Medidas de bondad de las reglas

## Soporte de la regla

Es la fracción de instancias que contienen al antecedente y a la consecuencia

$$\text{Supp}(A \rightarrow B) = \text{Supp}(A \cup B)$$

## Confianza de la regla

Fracción de instancias en las que aparece A que también incluyen a B; esto es, la confianza mide con qué frecuencia aparece B en las instancias que incluyen A

$$\text{Conf}(A \rightarrow B) = \frac{\text{Supp}(A \cup B)}{\text{Supp}(A)}$$

# Medidas de bondad de las reglas

ID Cesta	Cesta
001	Lechuga, Embutido, Cerveza, Pan de Molde
010	Cerveza, Pañales, Queso, Lechuga, Champú
011	Pizza, Refresco, Palomitas
100	Pañales, Jabón, Champú

$$\text{Supp}(\text{Cerveza} \rightarrow \text{Panales}) = \frac{1}{4} = 0.25$$

$$\text{Conf}(\text{Cerveza} \rightarrow \text{Panales}) = \frac{1}{2} = 0.5$$

# Medidas de bondad de las reglas

ID Cesta	Cesta
001	Lechuga, Embutido, Cerveza, Pan de Molde
010	Cerveza, Pañales, Queso, Lechuga, Champú
011	Pizza, Refresco, Palomitas
100	Pañales, Jabón, Champú

$$\text{Supp}(\text{Cerveza} \rightarrow \text{Panales}) = \frac{1}{4} = 0.25$$

$$\text{Conf}(\text{Cerveza} \rightarrow \text{Panales}) = \frac{1}{2} = 0.5$$

$$\text{Supp}(\text{Lechuga} \rightarrow \text{Cerveza}) = \frac{2}{4} = 0.5$$

$$\text{Conf}(\text{Lechuga} \rightarrow \text{Cerveza}) = \frac{2}{2} = 1$$

# Medidas de bondad de las reglas

- ❑ La confianza de una regla mide su **calidad**
- ❑ El soporte mide la **cantidad** de tuplas que soportan la inducción
- ❑ Se suelen imponer dos umbrales:
  - Umbral de Soporte **minsup** (5% p.e)  
Toda regla  $A \rightarrow C$  es frecuente si  $\text{supp}(A \rightarrow C) \geq \text{minsup}$
  - Umbral de Confianza **minconf** (70% p.e)

# Fases

- **Extracción de reglas** con soporte y confianza mayores que los umbrales (minsupp y minconf)
  - Algoritmos de Minería de reglas: A priori y variantes
- **Interpretación de las reglas**
  - Otras medidas de calidad: factor de certeza etc..
  - Otros tipos de reglas más complejas que impliquen causalidad
  - Mecanismos de agrupamiento de items en conceptos más complejos (P.E. no leche pascual, sino leche o producto lácteos) (Reglas de asociación difusas)
  - Uso de otros tipos de conjuntos para generar reglas en lugar de itemset frecuentes (item sets cerrados)
  - Mecanismo de Minería de segundo nivel (agrupamiento de reglas) etc.

# Aplicaciones

## Colocación de productos en las estanterías de un supermercado

### Problema

Identificar artículos que muchos clientes compran a la vez

### Solucion

Procesar los datos de los terminales de punto de venta proporcionados por los escáneres de códigos de barras

### Ejemplos

Si un cliente compra pañales, es muy probable que compre cerveza (! cervezas colocadas al lado de los pañales en el super!)

Wal-Mart sabe que los clientes que compran muñecas Barbie tienen un 60% de probabilidad de comprar uno de los tres tipos de barras de chocolate...

# Aplicaciones

## Fecha de los exámenes del Grado en Informática

### Problema

El subdirector de Ordenación Docente quiere fijar la fecha de los exámenes de Febrero de manera que los alumnos dispongan del máximo tiempo posible para estudiar entre examen y examen

### Solución

Procesar los datos sobre alumnos y asignaturas escogidas buscando co-ocurrencias frecuentes y colocando los exámenes separados

# Aplicaciones

## Sistema de recomendaciones

### Problema

Amazon desea vender más libros a clientes que han realizado una compra en alguna ocasión

### Solución

Procesar los datos de compras de libros y sugerir aquellos libros co-ocurrentes con el libro comprado por el cliente

# Índice

- Concepto de regla de asociación
- **Algoritmo Apriori**
- Variaciones del algoritmo A priori
- Algoritmo FP-Growth
- Generación de reglas
- Evaluación de reglas

# Problema básico

Dado un conjunto de instancias, encontrar todas las reglas de asociación tales que:

- el soporte sea mayor o igual que un umbral mínimo de soporte, MinSupp

$$\text{Supp}(X \rightarrow Y) \geq \text{MinSupp}$$

- la confianza sea mayor o igual que un umbral mínimo de confianza, MinConf

$$\text{Conf}(X \rightarrow Y) \geq \text{MinConf}$$

# Solución ineficiente

Algoritmo por **fuerza bruta**:

1. Calcular todas las reglas posibles
2. Calcular el soporte y confianza de las reglas
3. Seleccionar las que cumplen los umbrales mínimos
4. Devolver las reglas seleccionadas

Eficiencia exponencial!



# Solución ineficiente

Por ejemplo,  
reglas derivadas de  
 $\{\text{pan}, \text{pañales}, \text{cerveza}\}$

ID	Cesta
001	Pan, leche, huevos
010	Pan, pañales, cerveza
011	Leche, pañales, cerveza
100	Pan, leche, pañales, cerveza
101	Pan, leche, huevos, cerveza

- $\{\text{pan}\} \rightarrow \{\text{pañales, cerveza}\}$ , supp=0.4, conf=2/4=0.5
- $\{\text{pañales}\} \rightarrow \{\text{pan, cerveza}\}$ , supp=0.4, conf=2/3=0.66
- $\{\text{cerveza}\} \rightarrow \{\text{pan, pañales}\}$ , supp=0.4, conf=2/4=0.5
- $\{\text{pan, pañales}\} \rightarrow \{\text{cerveza}\}$ , supp=0.4, conf=2/2=1
- $\{\text{pan, cerveza}\} \rightarrow \{\text{pañales}\}$ , supp=0.4, conf=2/3=0.66
- $\{\text{pañales, cerveza}\} \rightarrow \{\text{pan}\}$ , supp=0.4, conf=2/3=0.66

# Cálculo eficiente de reglas

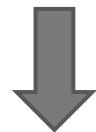
En el ejemplo anterior,

- ❑ Todas las reglas son particiones del itemset  
    {pan, pañales, cerveza}
- ❑ Entonces, todas **las reglas tienen el mismo soporte** (el soporte del itemset)
- ❑ La confianza puede variar

# Cálculo eficiente de reglas

En el ejemplo anterior,

- ❑ Todas las reglas son particiones del itemset  
    {pan, pañales, cerveza}
- ❑ Entonces, todas **las reglas tienen el mismo soporte** (el soporte del itemset)
- ❑ La confianza puede variar



Dividir el cálculo en buscar itemsets con soporte mínimo y después en buscar la confianza mínima

# Cálculo eficiente de reglas

Solución en dos etapas:

## 1. Generación de itemsets frecuentes

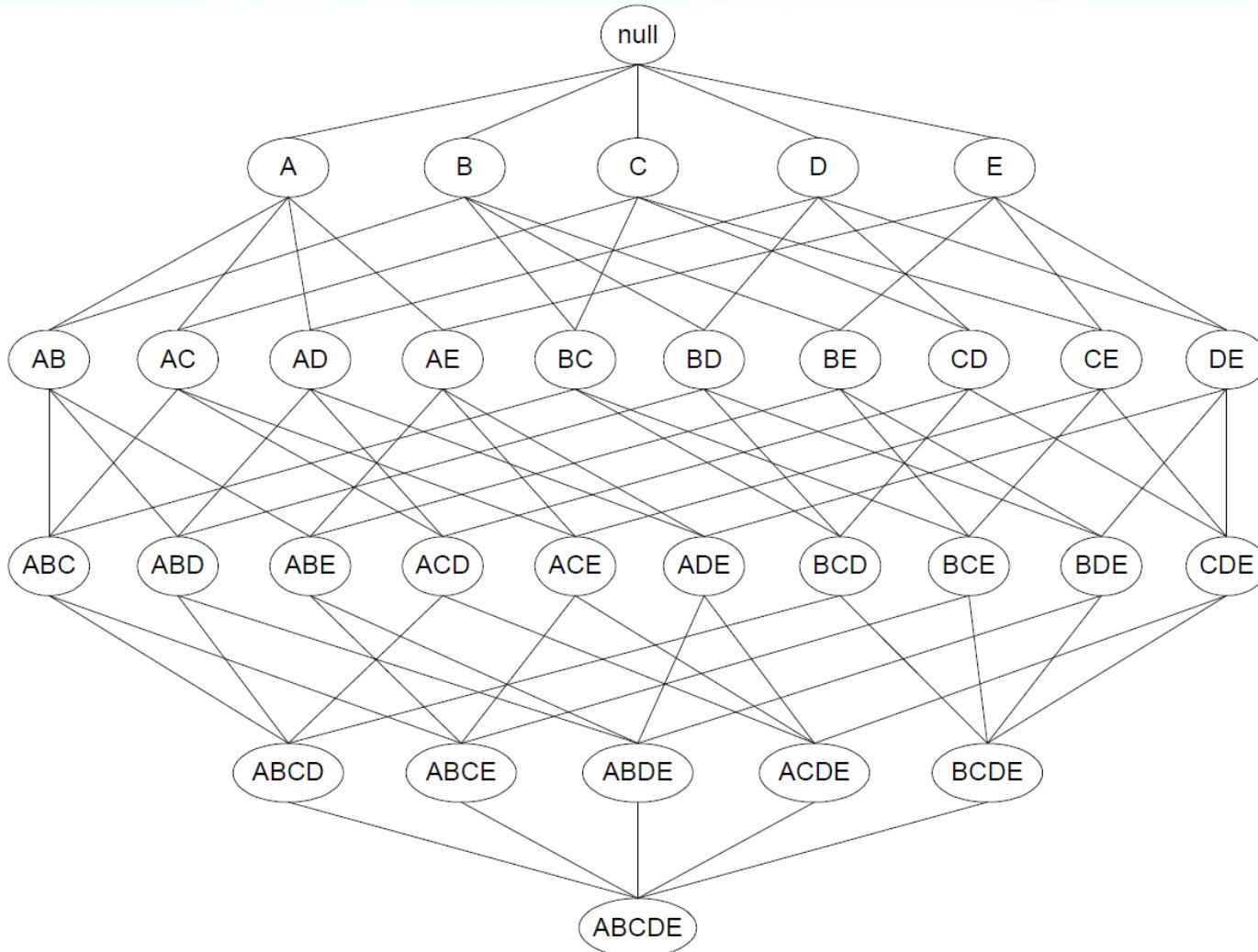
Identificar los itemsets con soporte mayor que MinSupp

## 2. Generacion de reglas de asociacion

Obtener reglas de asociación con una confianza mayor que MinConf a partir de cada itemset frecuente, donde cada regla es una partición binaria del itemset

# Generación itemsets frecuentes

Calcularlos exhaustivamente sigue siendo poco eficiente



# Generación itemsets frecuentes

Calcularlos exhaustivamente sigue siendo poco eficiente

Para d items distintos tenemos:

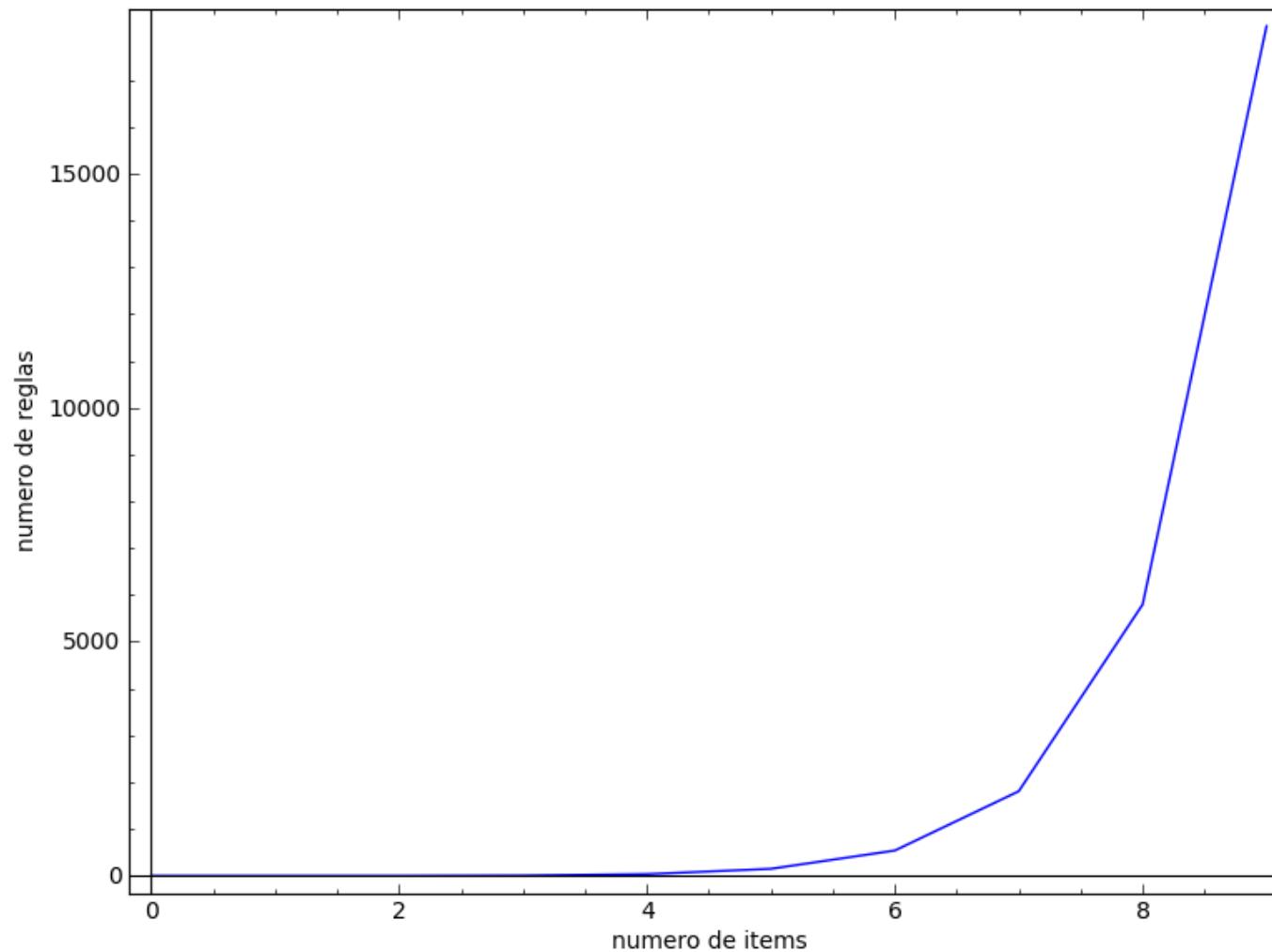
- $\sum_{i=1}^d \binom{d}{i} = 2^d - 1$  itemsets no vacíos

Y luego...

- $\sum_{i=1}^d \binom{d}{i} \sum_{j=1}^{j=i-1} \binom{i}{j} = 3^d - 2^{d+1} + 1$  reglas

# Generación itemsets frecuentes

Calcularlos exhaustivamente sigue siendo poco eficiente



# Generación de itemsets frecuentes

Estrategias para generar itemsets frecuentes:

## Reducir el numero de candidatos (poda)

- Algoritmo Apriori
- 
- Algoritmo básico**

## Reducir el numero de instancias

- Conforme aumenta el tamaño del itemset
- Muestreo

## Reducir el numero de comparaciones

- Uso de estructuras de datos eficientes para almacenar los candidatos o las transacciones (tablas hash)
- Algoritmo Eclat, Declat (para cálculo del soporte)
- Algoritmo FP-Growth (estructura de árbol)

# Generación de itemsets frecuentes

## Propiedad Apriori

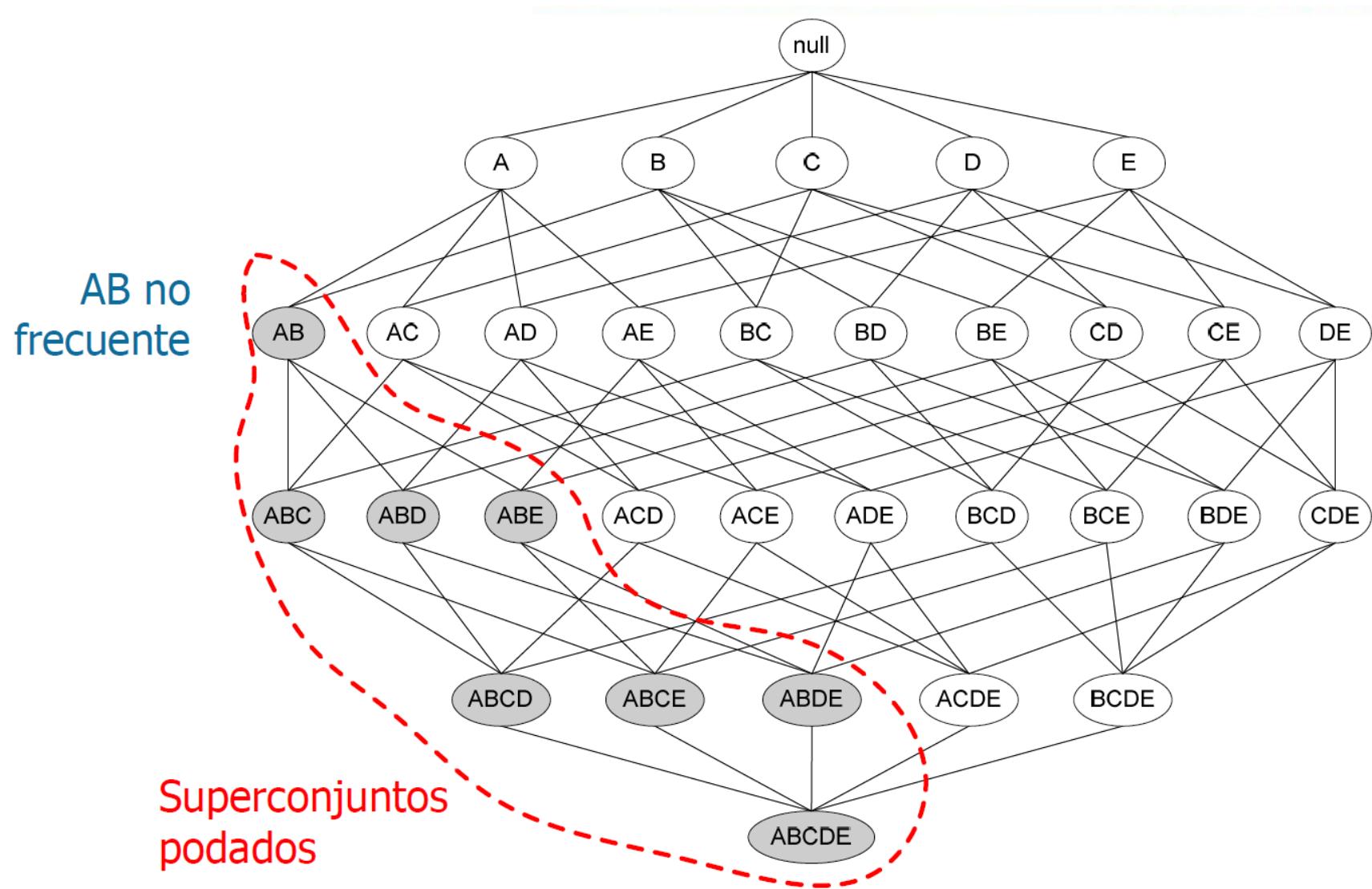
- ❑ Si un itemset es frecuente, todo subconjunto suyo también es frecuente
- ❑ O, equivalentemente, si un itemset no es frecuente, entonces todo itemset que lo contenga tampoco es frecuente

Que se deduce claramente de la propiedad

$$A \subseteq B \implies \text{Supp}(A) \geq \text{Supp}(B)$$

Antimonotonía

# Propiedad Apriori



# Algoritmo Apriori

---

**Algorithm** Algoritmo generador Apriori

---

**Input:**  $\mathcal{D}$  conjunto de transacciones,  $d$  soporte mínimo

**Output:**  $\mathcal{L}$  conjunto de itemsets frecuentes

- 1:  $L_1 \leftarrow \{1\text{-itemsets } A \text{ con } \text{Supp}(A) \geq d\}$
  - 2:  $k \leftarrow 2$
  - 3: **while**  $L_{k-1} \neq \emptyset$  **do**
  - 4:      $C_k \leftarrow \{A \cup B \text{ donde } A, B \in L_{k-1}, |A \cup B| = k\}$
  - 5:      $C_k \leftarrow \{D \in C_k \text{ tales que } C \subset D, |C| = k - 1 \Rightarrow C \in L_{k-1}\}$
  - 6:      $L_k \leftarrow \{c \in C_k \text{ con } \text{Supp}(c) \geq d\}$
  - 7:      $k = k + 1$
  - 8:  $\mathcal{L} = \bigcup_{k \geq 1} L_k$
  - 9: **return**  $\mathcal{L}$
-

# Ejemplo (MinSupp=2)

Cesta compra

ID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

# Ejemplo (MinSupp=2)

Cesta compra

ID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

Cálculo  
soporte

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

# Ejemplo (MinSupp=2)

Cesta compra

ID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

Cálculo  
soporte

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

→

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

# Ejemplo (MinSupp=2)

Cesta compra

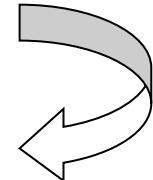
ID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

Cálculo  
soporte

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

→

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3



Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

# Ejemplo (MinSupp=2)

Cesta compra

ID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

Cálculo  
soporte

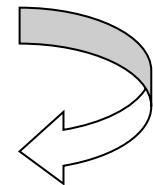
Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

Cálculo  
soporte

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}



# Ejemplo (MinSupp=2)

Cesta compra

ID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

Cálculo  
soporte

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3



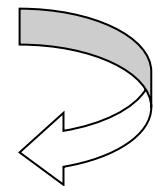
Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2



Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

Cálculo  
soporte

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}



# Ejemplo (MinSupp=2)

Cesta compra

ID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

Cálculo  
soporte

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3



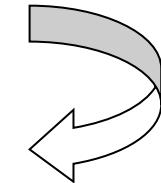
Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2



Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

Cálculo  
soporte

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}



Itemset
{B, C, E}

# Ejemplo (MinSupp=2)

Cesta compra

ID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

Cálculo  
soporte

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3



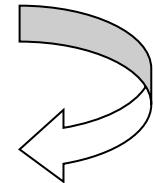
Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2



Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

Cálculo  
soporte

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}



Itemset
{B, C, E}

Cálculo  
soporte

Itemset	sup
{B, C, E}	2

# Ejemplo (MinSupp=2)

Cesta compra

ID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

Itemsets frecuentes

Itemset	sup
{B, C, E}	2

# Índice

- Concepto de regla de asociación
- Algoritmo Apriori
- Variaciones del algoritmo A priori
- Algoritmo FP-Growth
- Evaluación de reglas
- Reglas difusas
- Otros aspectos

# Apriori particionado

---

**Algorithm** Algoritmo Apriori particionado

---

**Input:**  $\mathcal{D}$  conjunto de transacciones,  $\{\mathcal{D}_i\}_{i=1,\dots,n}$  partición de  $\mathcal{D}$ ,  $d$  soporte mínimo

**Output:**  $\mathcal{L}$  conjunto de itemset frecuentes

- 1: **for**  $1 \leq i \leq n$  **do**
  - 2:      $T_j^i \leftarrow \text{GEN-APRIORI}(\mathcal{D}_i, d)$  para  $j \leq \#\text{items}$
  - 3: **for**  $1 \leq j \leq \#\text{items}$  **do**
  - 4:      $\mathcal{T}_j = \cup_{i=1}^n T_i^j$
  - 5:  $\mathcal{T} = \cup_j \mathcal{T}_j$
  - 6:  $\mathcal{L} = \{c \in \mathcal{T} \text{ tales que } \text{Supp}(c) \geq d\}$
  - 7: **return**  $\mathcal{L}$
-

# Sampling

- ❑ Se escoge una **muestra aleatoria** en el conjunto global de instancias y en la muestra se busca conjuntos de elementos frecuentes
- ❑ Por lo general, la muestra es lo suficientemente grande como para caber en la memoria principal
- ❑ Estos conjuntos de elementos frecuentes se llaman conjuntos frecuentes de muestra (**sample frequent itemsets**)

# Sampling

- ❑ Se pierde exactitud, pero se gana velocidad
- ❑ Se le suele exigir un soporte mínimo a la muestra para asegurar que no se pierden itemsets frecuentes
- ❑ Se pueden realizar varios muestreos y así conseguir mejor exactitud

# Índice

- Concepto de regla de asociación
- Algoritmo Apriori
- Variaciones del algoritmo A priori
- **Algoritmo FP-Growth**
- Generación de reglas
- Evaluación de reglas
- Reglas difusas
- Otros aspectos

# FP-Growth Algorithm

Basado en tres operaciones:

- ❑ Cálculo del FP-Tree
  - Almacenamiento de la base de datos en un árbol
- ❑ Proyección del FP-Tree a un ítem
- ❑ Eliminación de ítems no frecuentes

Es un algoritmo recursivo

# Construcción del FP-Tree

	Items
1	ABDE
2	BCE
3	ABDE
4	ABCE
5	ABCDE
6	BCD

# Construcción del FP-Tree

## Cálculo del soporte de los items

	Items
1	ABDE
2	BCE
3	ABDE
4	ABCE
5	ABCDE
6	BCD

A(4)

B(6)

C(4)

D(4)

E(5)

# Construcción del FP-Tree

**Reordenamos en orden decreciente**

B(6) > E(5) > A(4) > C(4) > D(4)

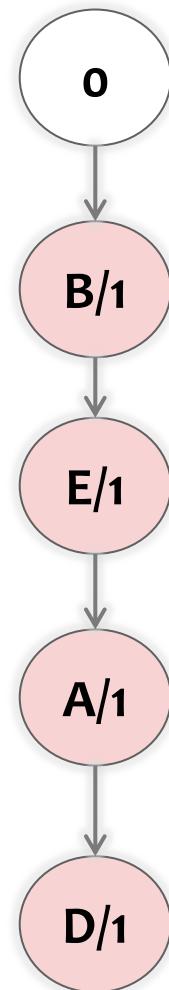
	Items
1	BEAD
2	BEC
3	BEAD
4	BEAC
5	BEACD
6	BCD

# Construcción del FP-Tree

Añadimos transacción 1

B(6) > E(5) > A(4) > C(4) > D(4)

	Items
1	<b>BEAD</b>
2	BEC
3	BEAD
4	BEAC
5	BEACD
6	BCD

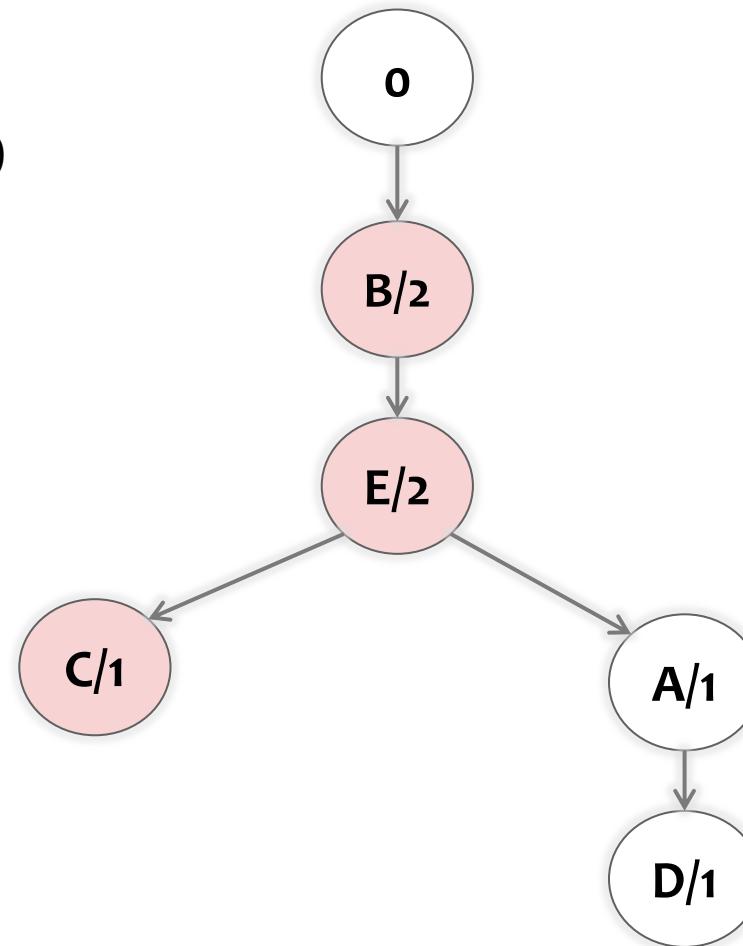


# Construcción del FP-Tree

Añadimos transacción 2

B(6) > E(5) > A(4) > C(4) > D(4)

	Items
1	BEAD
2	<b>BEC</b>
3	BEAD
4	BEAC
5	BEACD
6	BCD

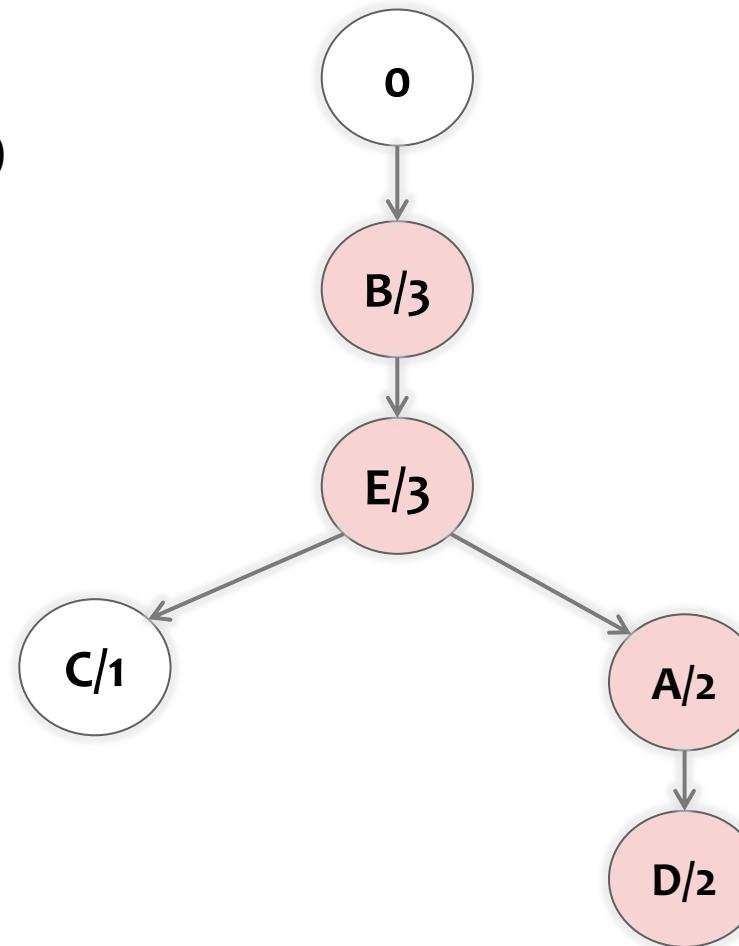


# Construcción del FP-Tree

Añadimos transacción 3

B(6) > E(5) > A(4) > C(4) > D(4)

	Items
1	BEAD
2	BEC
3	<b>BEAD</b>
4	BEAC
5	BEACD
6	BCD

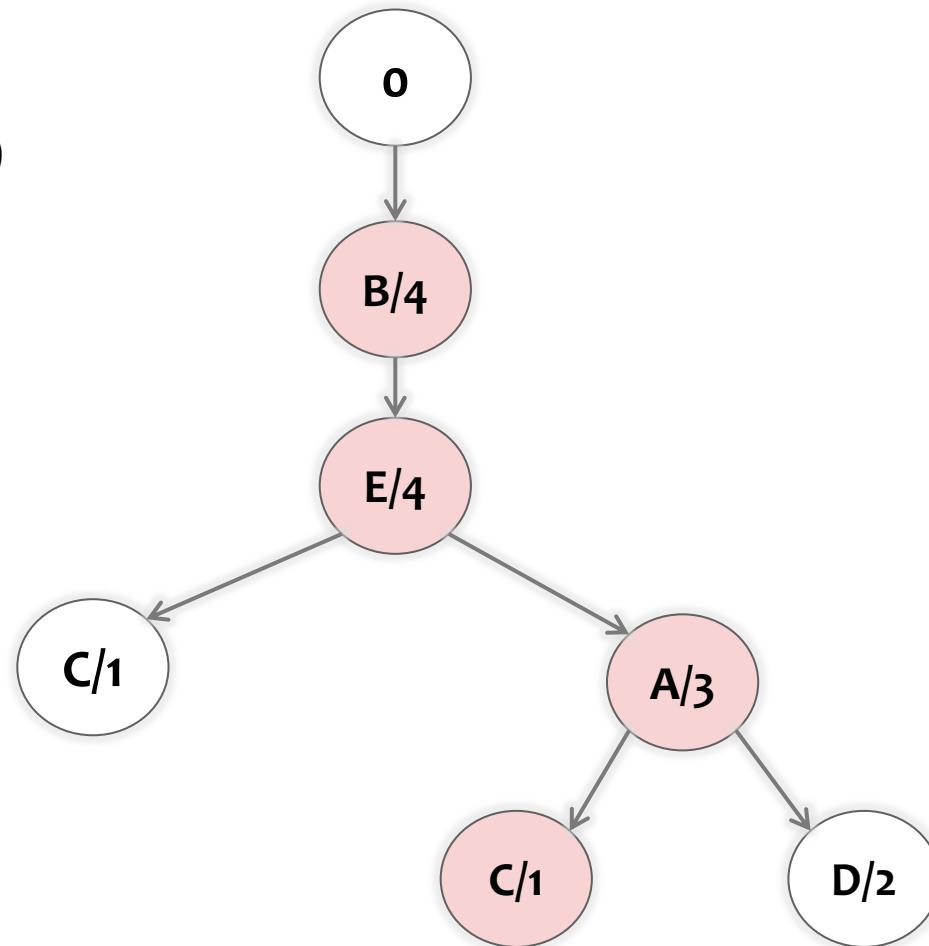


# Construcción del FP-Tree

Añadimos transacción 4

B(6) > E(5) > A(4) > C(4) > D(4)

	Items
1	BEAD
2	BEC
3	BEAD
4	<b>BEAC</b>
5	BEACD
6	BCD

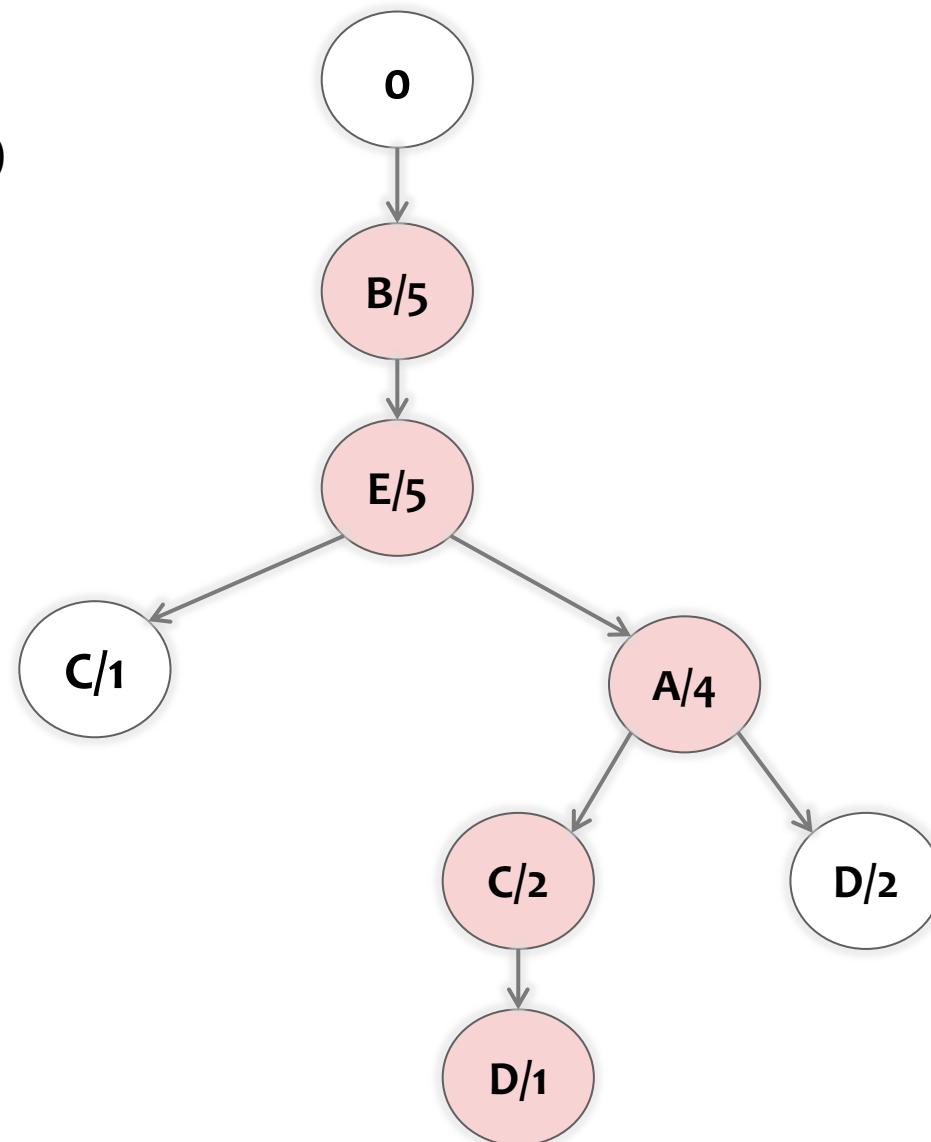


# Construcción del FP-Tree

Añadimos transacción 5

B(6) > E(5) > A(4) > C(4) > D(4)

	Items
1	BEAD
2	BEC
3	BEAD
4	BEAC
5	<b>BEACD</b>
6	BCD

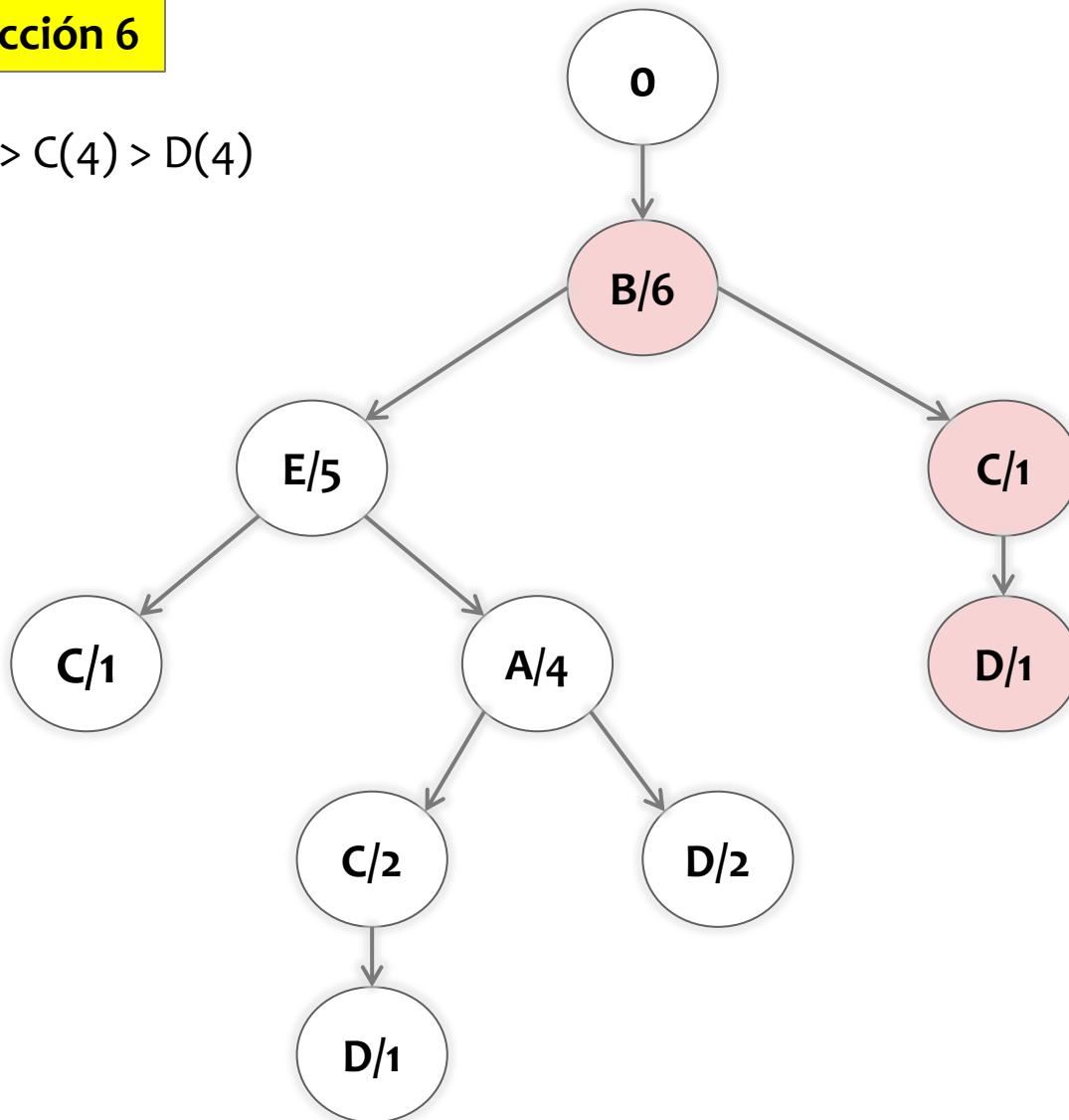


# Construcción del FP-Tree

Añadimos transacción 6

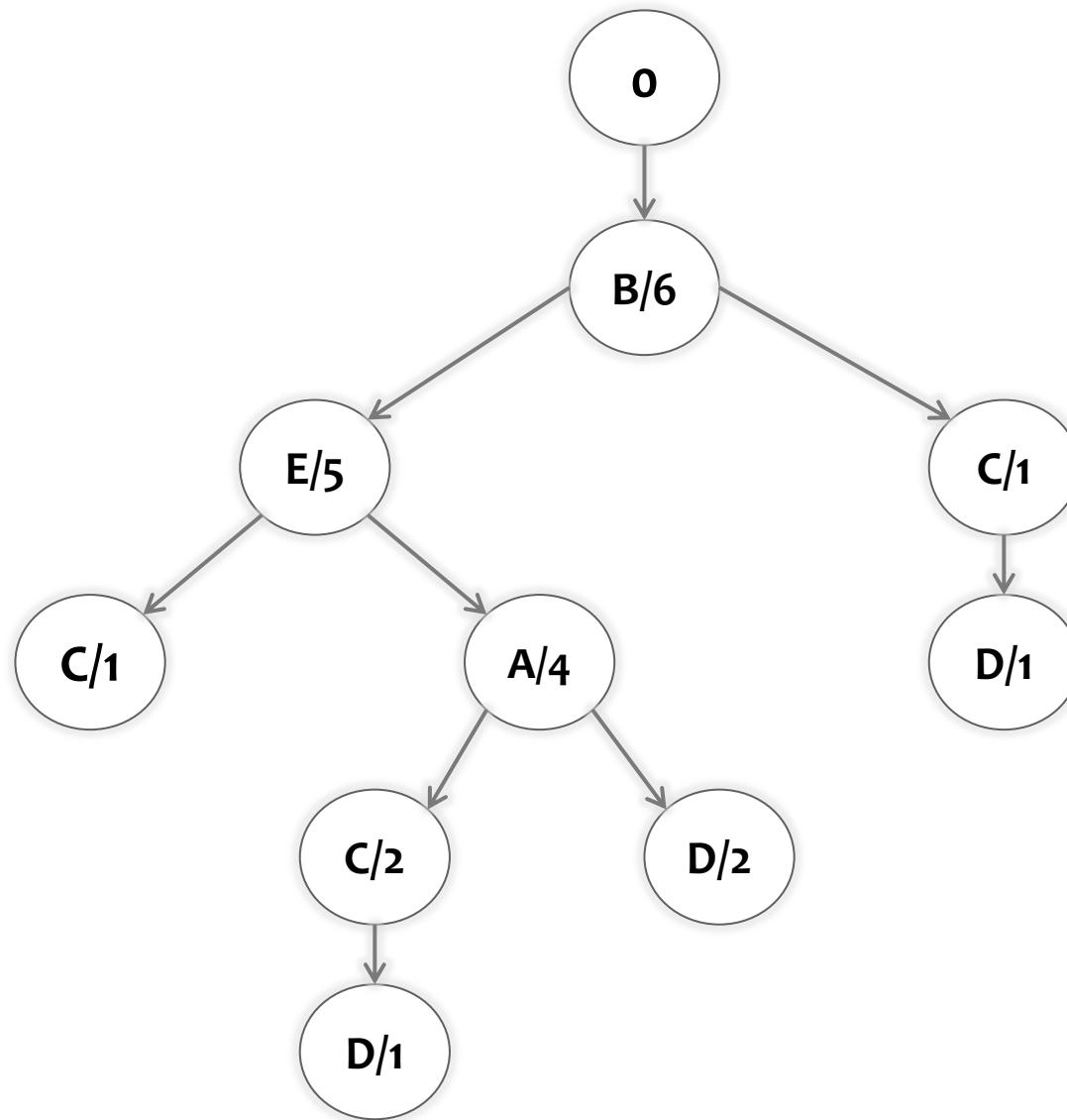
B(6) > E(5) > A(4) > C(4) > D(4)

	Items
1	BEAD
2	BEC
3	BEAD
4	BEAC
5	BEACD
6	<b>BCD</b>



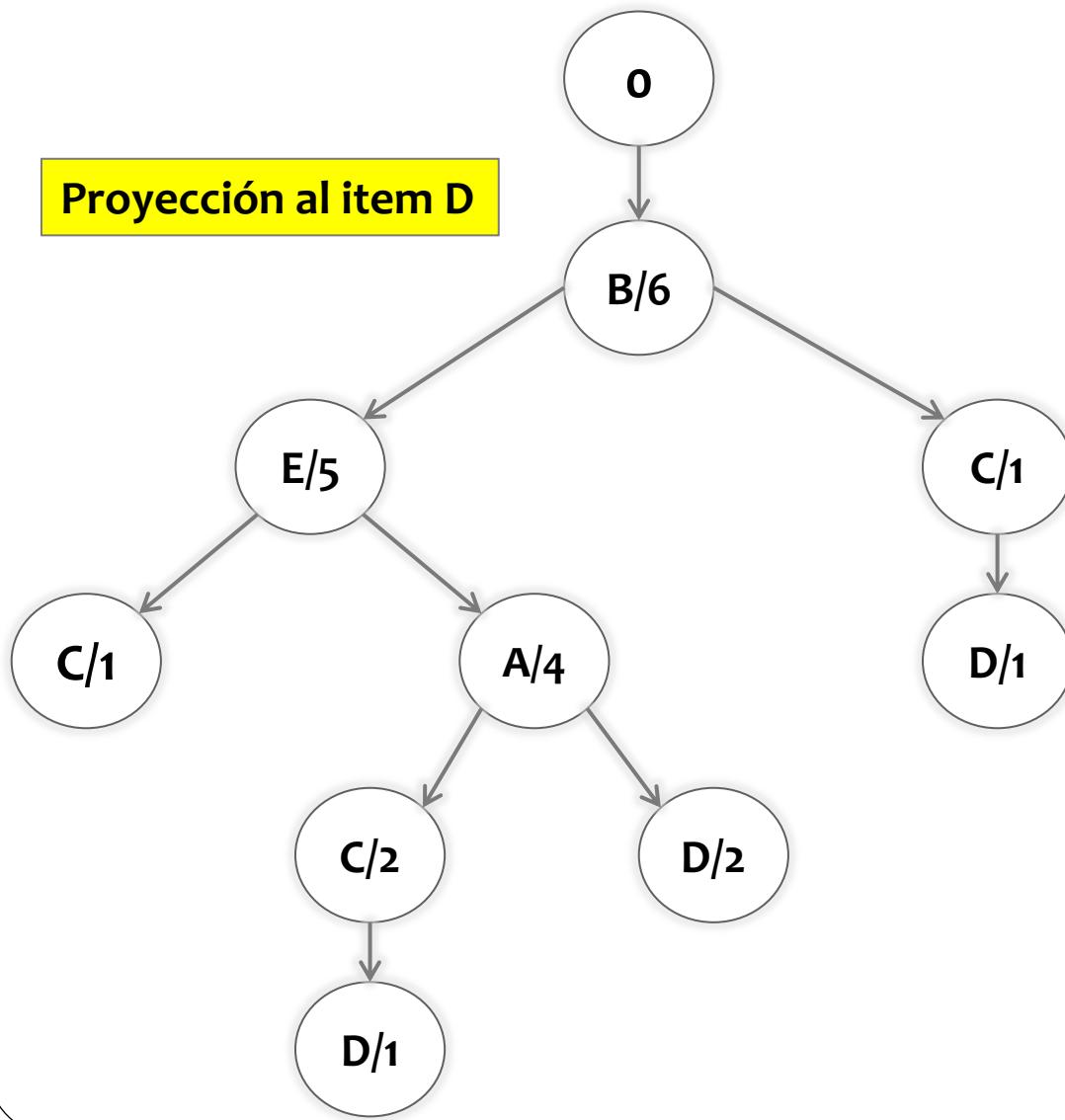
# Construcción del FP-Tree

	Items
1	BEAD
2	BEC
3	BEAD
4	BEAC
5	BEACD
6	BCD

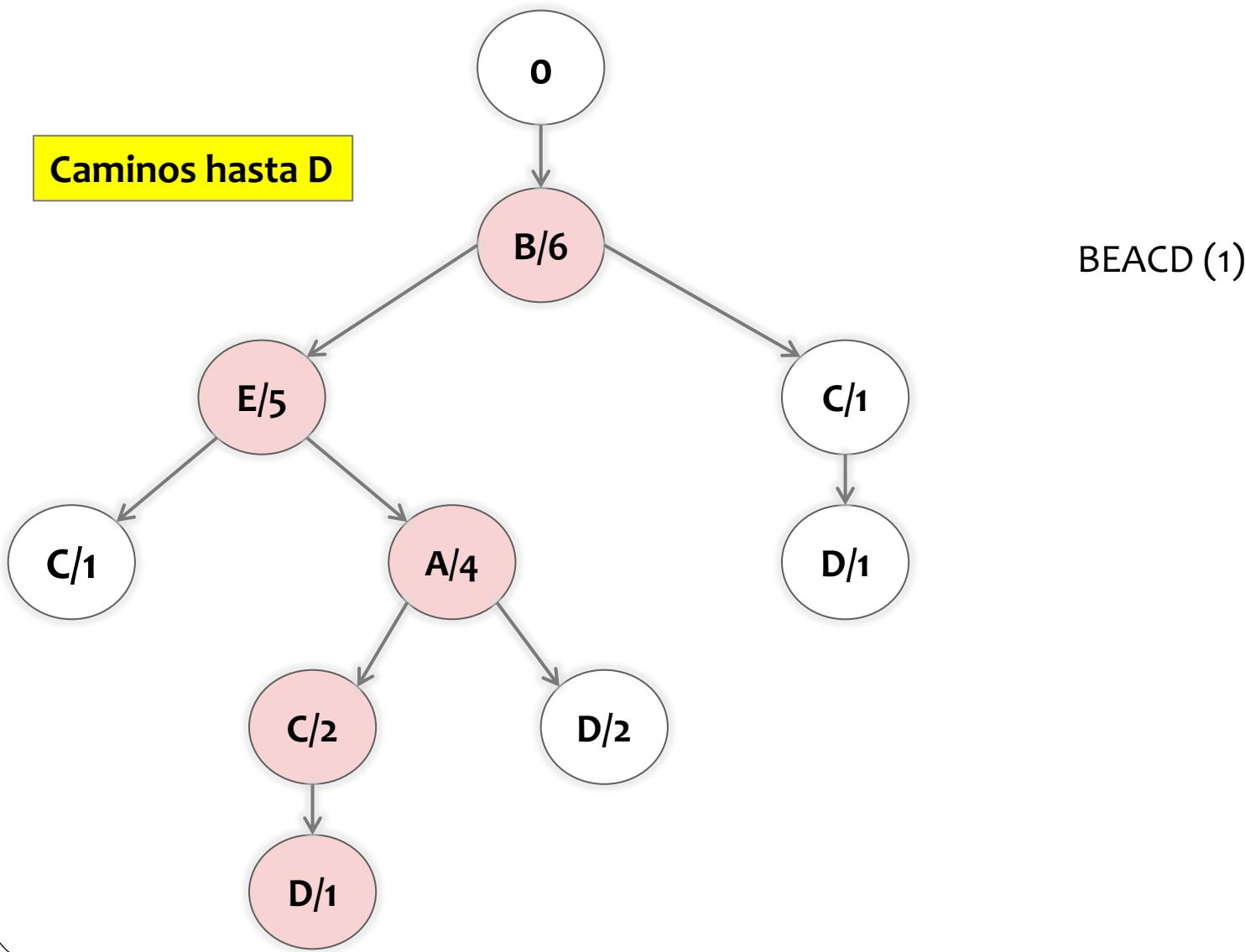


# Proyección del FP-Tree

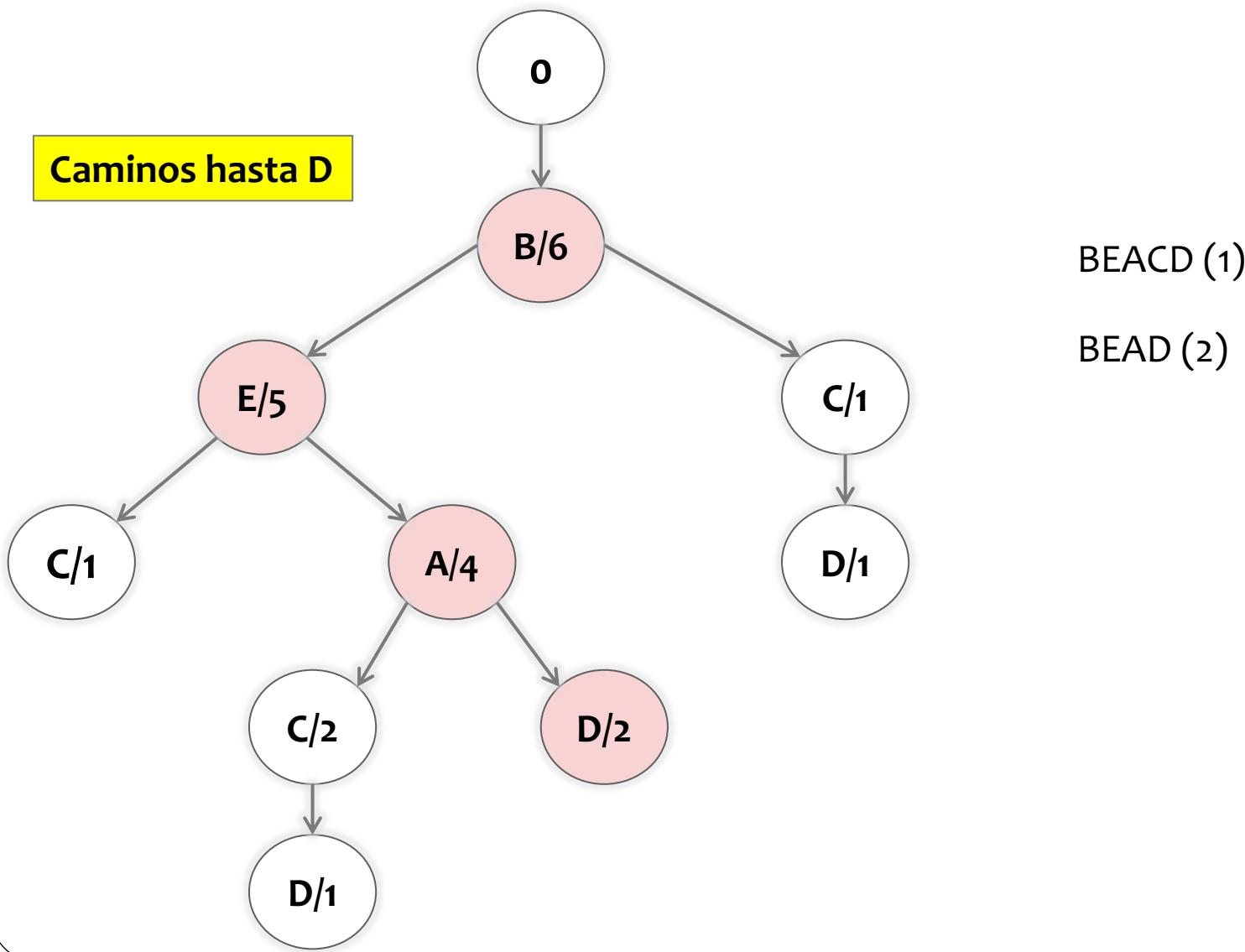
Proyección al ítem D



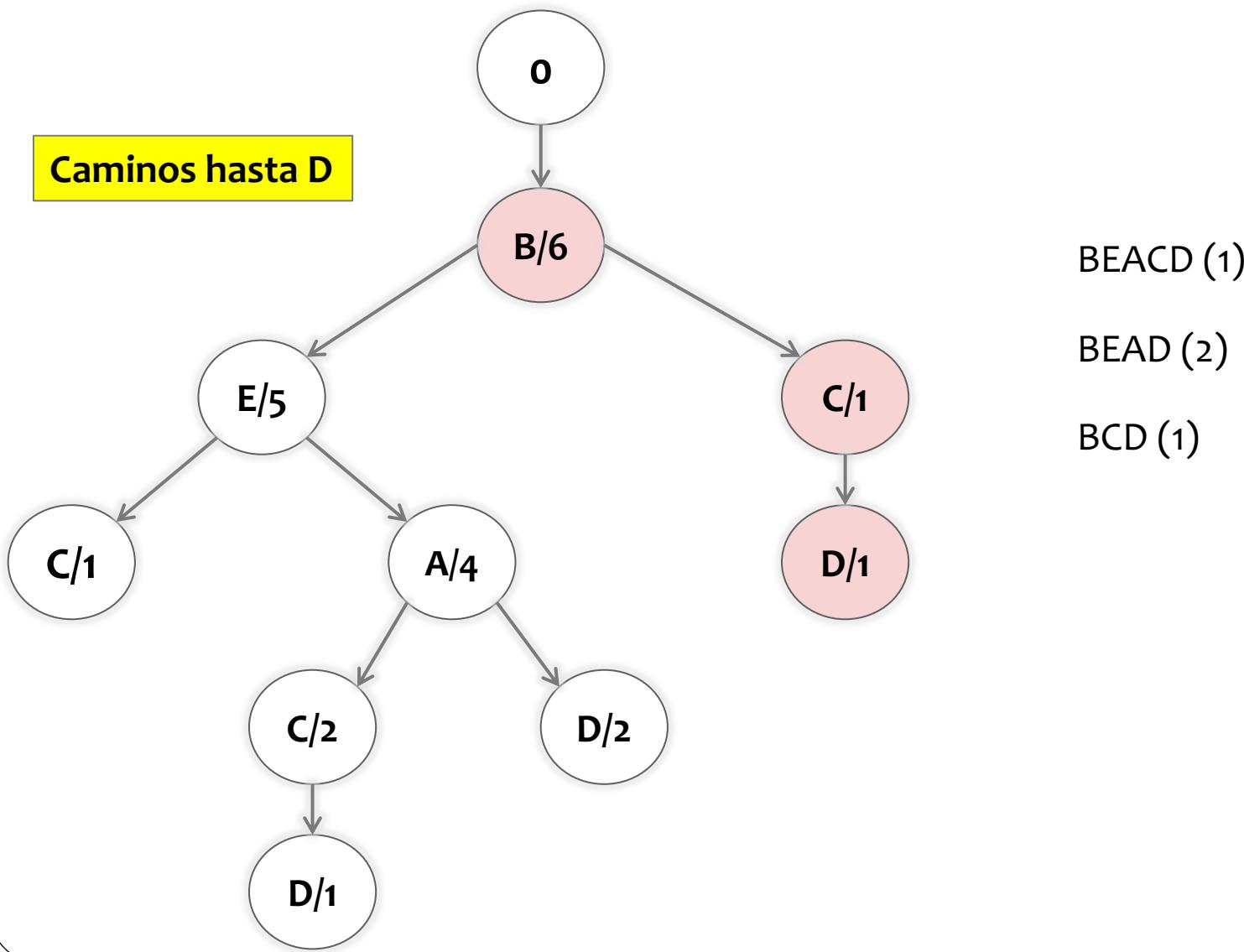
# Proyección del FP-Tree



# Proyección del FP-Tree



# Proyección del FP-Tree



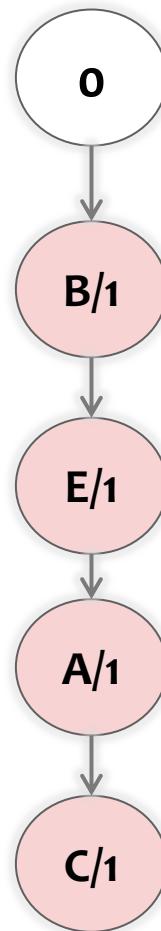
# Proyección del FP-Tree

FP-Tree de los caminos

BEACD (1)

BEAD (2)

BCD (1)



# Proyección del FP-Tree

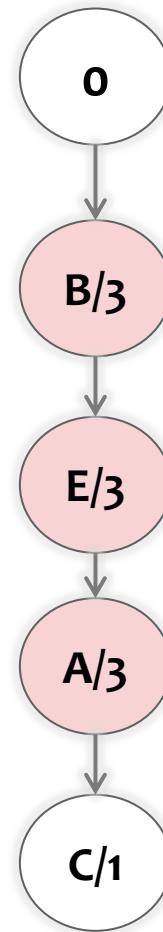
FP-Tree de los caminos

BEACD (1)

**BEAD (2)**

BCD (1)

... se suman  
dos...



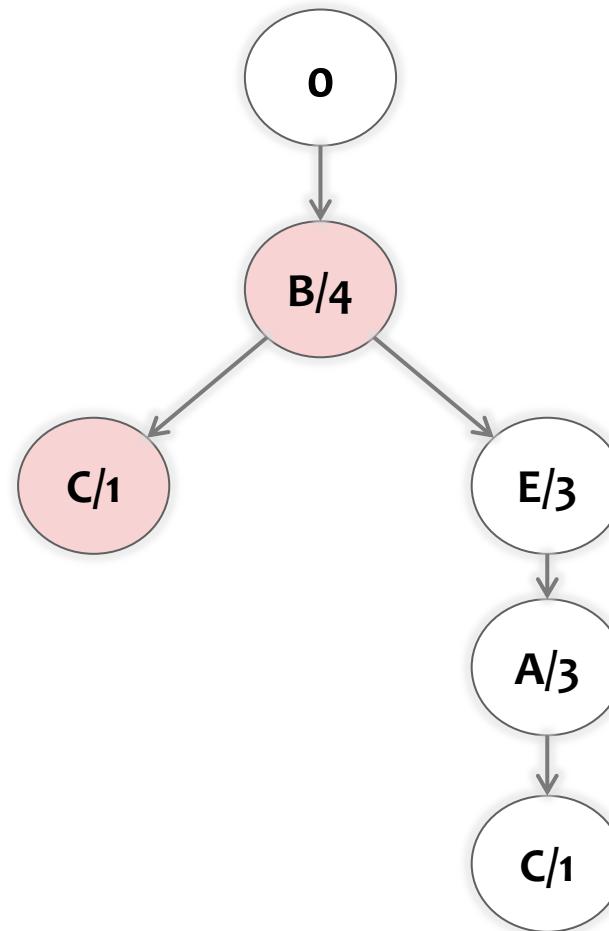
# Proyección del FP-Tree

FP-Tree de los caminos

BEACD (1)

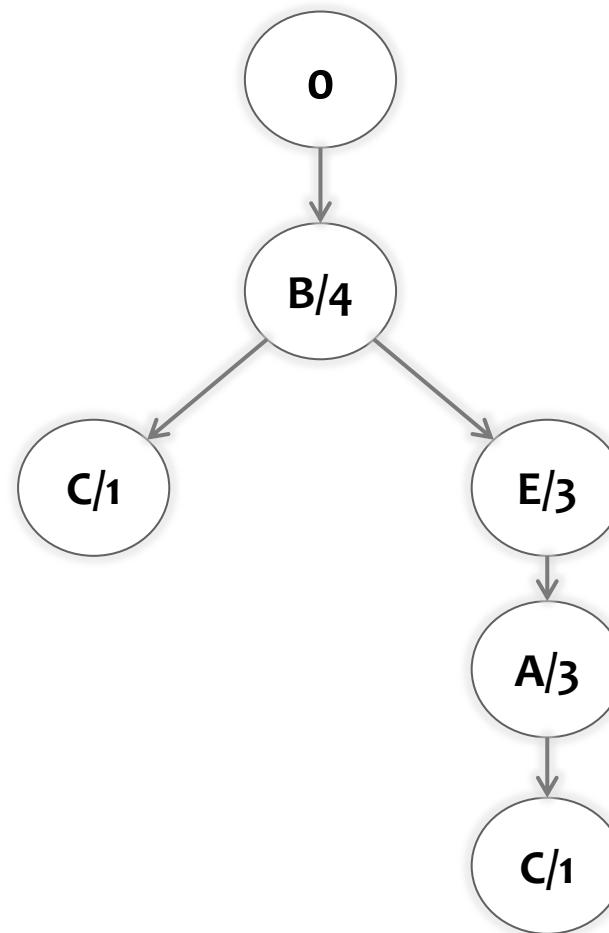
BEAD (2)

**BCD (1)**



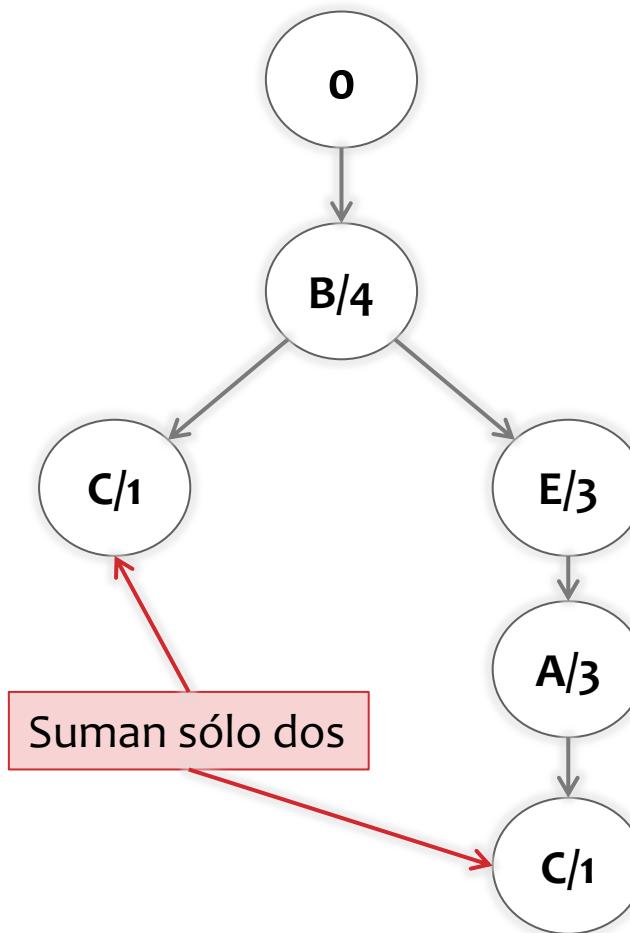
# Eliminación de items infrecuentes

Supongamos minsupp=3



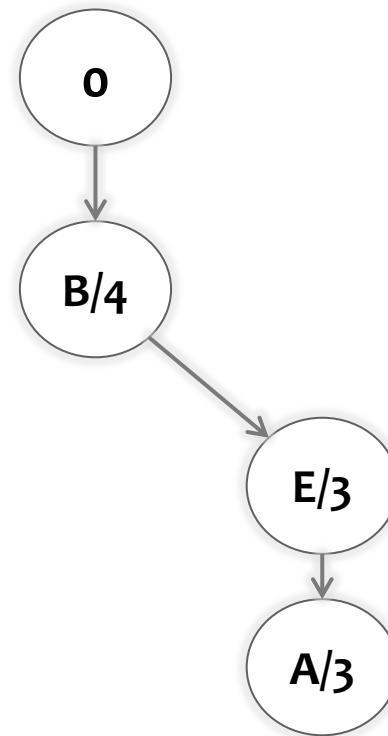
# Eliminación de items infrecuentes

Supongamos minsupp=3



# Eliminación de items infrecuentes

Supongamos  $\text{minsupp}=3$



# FP-Growth algorithm

---

## ALGORITHM 8.5. Algorithm FP GROWTH

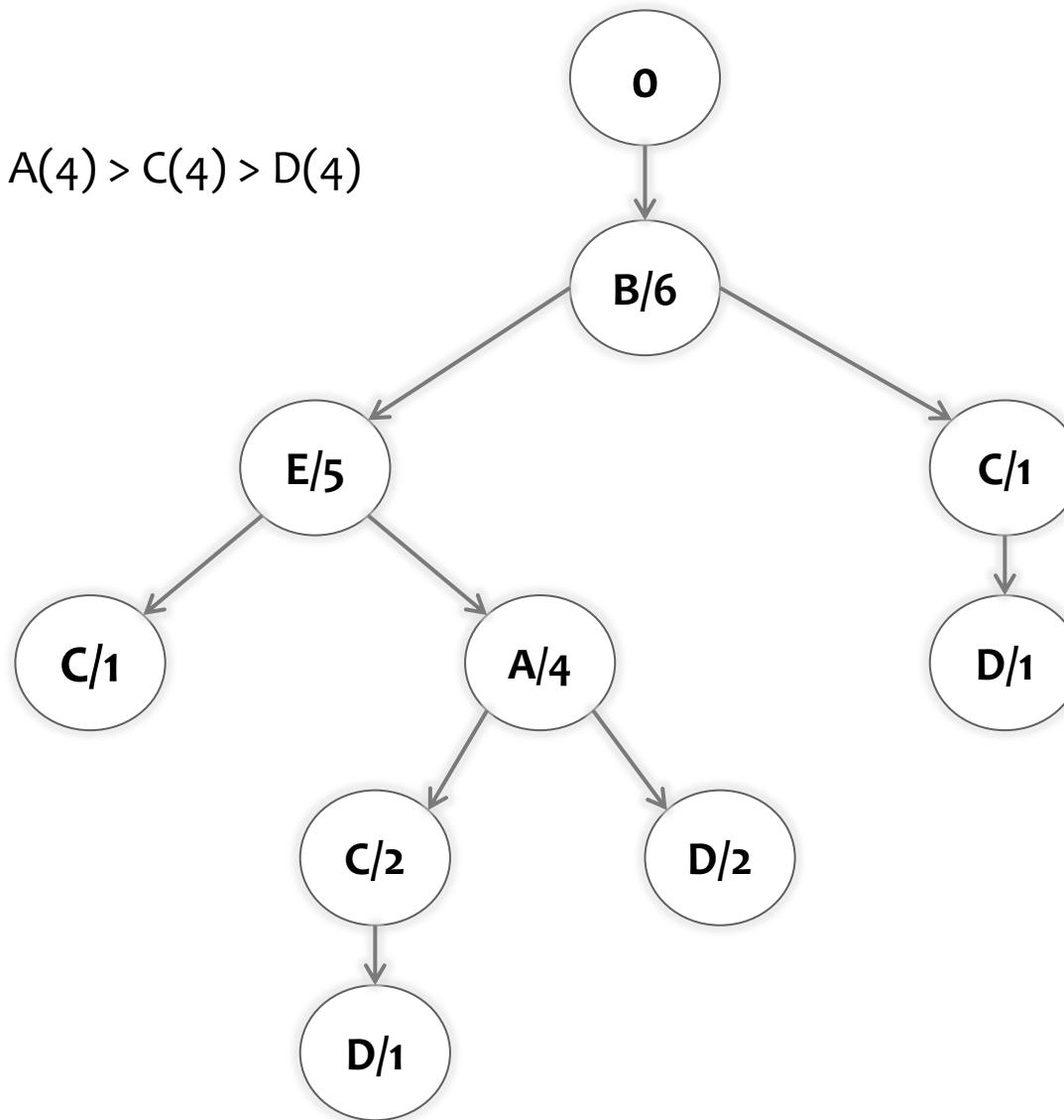
---

```
// Initial Call:  $R \leftarrow \text{FP-tree}(\mathbf{D})$ ,  $P \leftarrow \emptyset$ ,  $\mathcal{F} \leftarrow \emptyset$ 
FPGROWTH ( $R, P, \mathcal{F}, \text{minsup}$ ):
1 Remove infrequent items from  $R$ 
2 if  $\text{ISPATH}(R)$  then // insert subsets of  $R$  into  $\mathcal{F}$ 
3   foreach  $Y \subseteq R$  do
4      $X \leftarrow P \cup Y$ 
5      $\text{sup}(X) \leftarrow \min_{x \in Y} \{\text{cnt}(x)\}$  ← Valor del nodo
6      $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, \text{sup}(X))\}$ 
7 else // process projected FP-trees for each frequent item  $i$ 
8   foreach  $i \in R$  in increasing order of  $\text{sup}(i)$  do
9      $X \leftarrow P \cup \{i\}$ 
10     $\text{sup}(X) \leftarrow \text{sup}(i)$  // sum of  $\text{cnt}(i)$  for all nodes labeled  $i$ 
11     $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, \text{sup}(X))\}$ 
12     $R_X \leftarrow \emptyset$  // projected FP-tree for  $X$ 
13    foreach  $\text{path} \in \text{PATHFROMROOT}(i)$  do ← Cálculo de la proyección del FP-Tree
14       $\text{cnt}(i) \leftarrow \text{count of } i \text{ in } \text{path}$ 
15      Insert  $\text{path}$ , excluding  $i$ , into FP-tree  $R_X$  with count  $\text{cnt}(i)$ 
16    if  $R_X \neq \emptyset$  then FPGROWTH ( $R_X, X, \mathcal{F}, \text{minsup}$ )
```

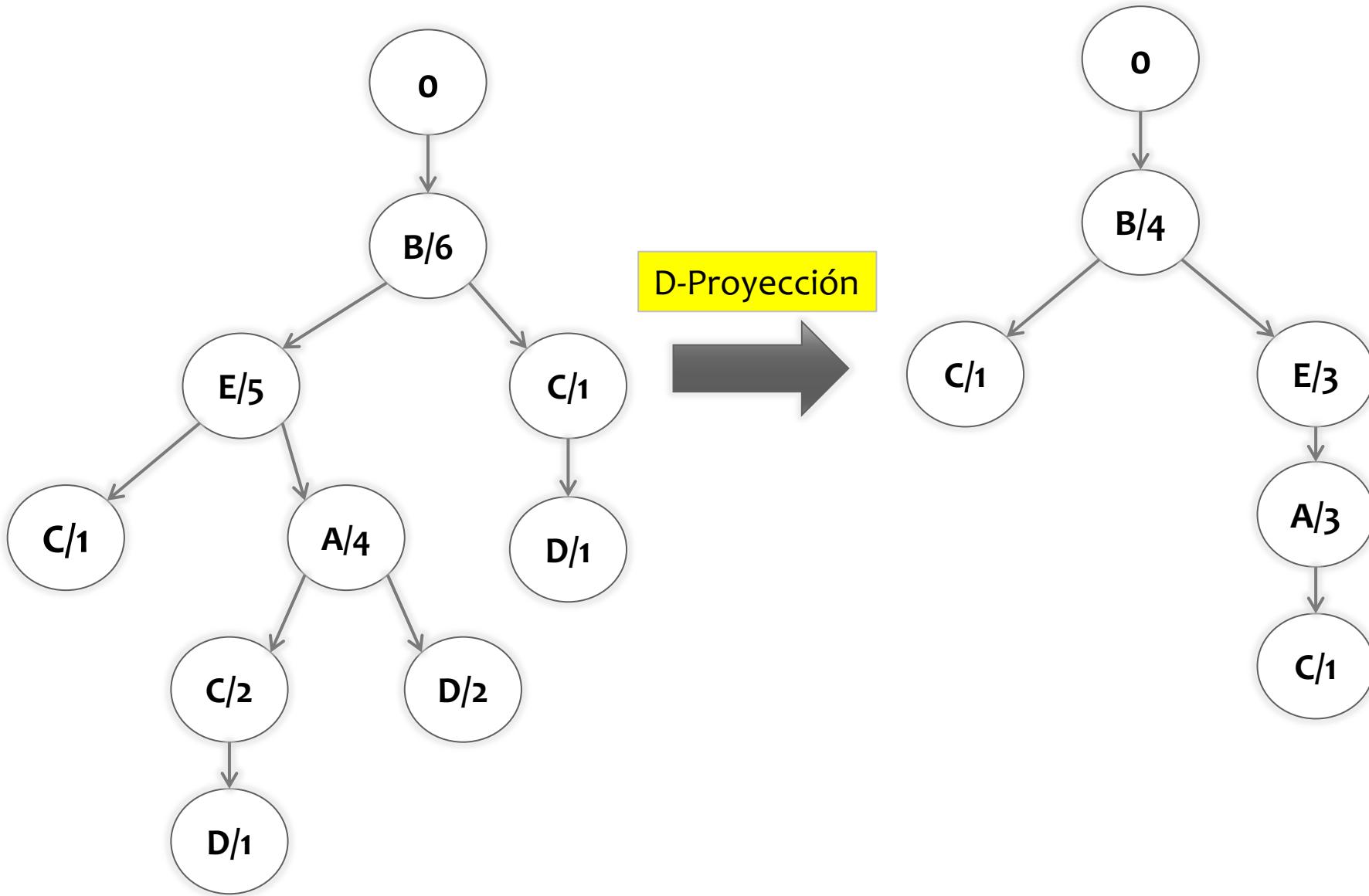
# Ejemplo (minsupp=3)

B(6) > E(5) > A(4) > C(4) > D(4)

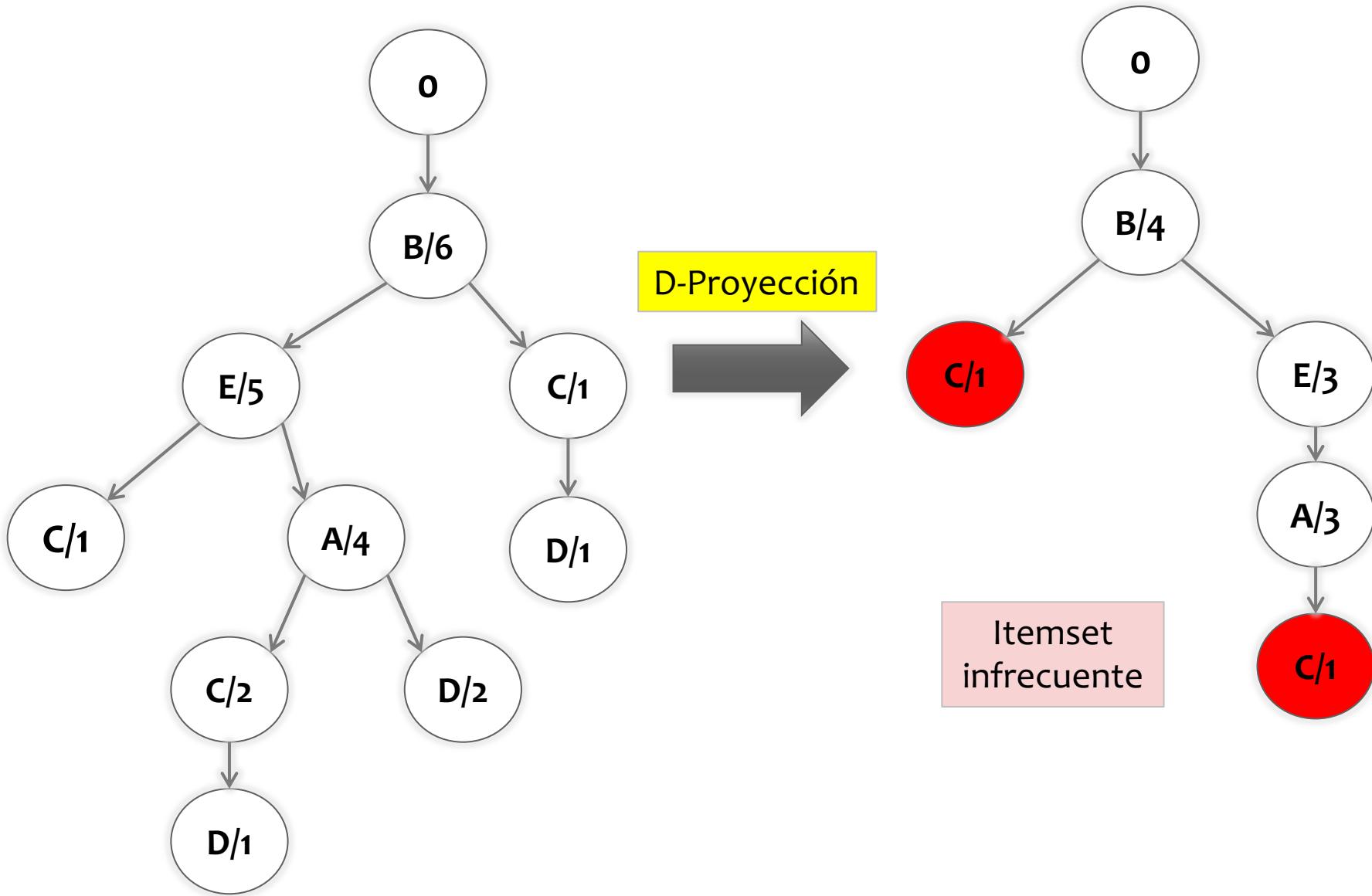
	Items
1	BEAD
2	BEC
3	BEAD
4	BEAC
5	BEACD
6	BCD



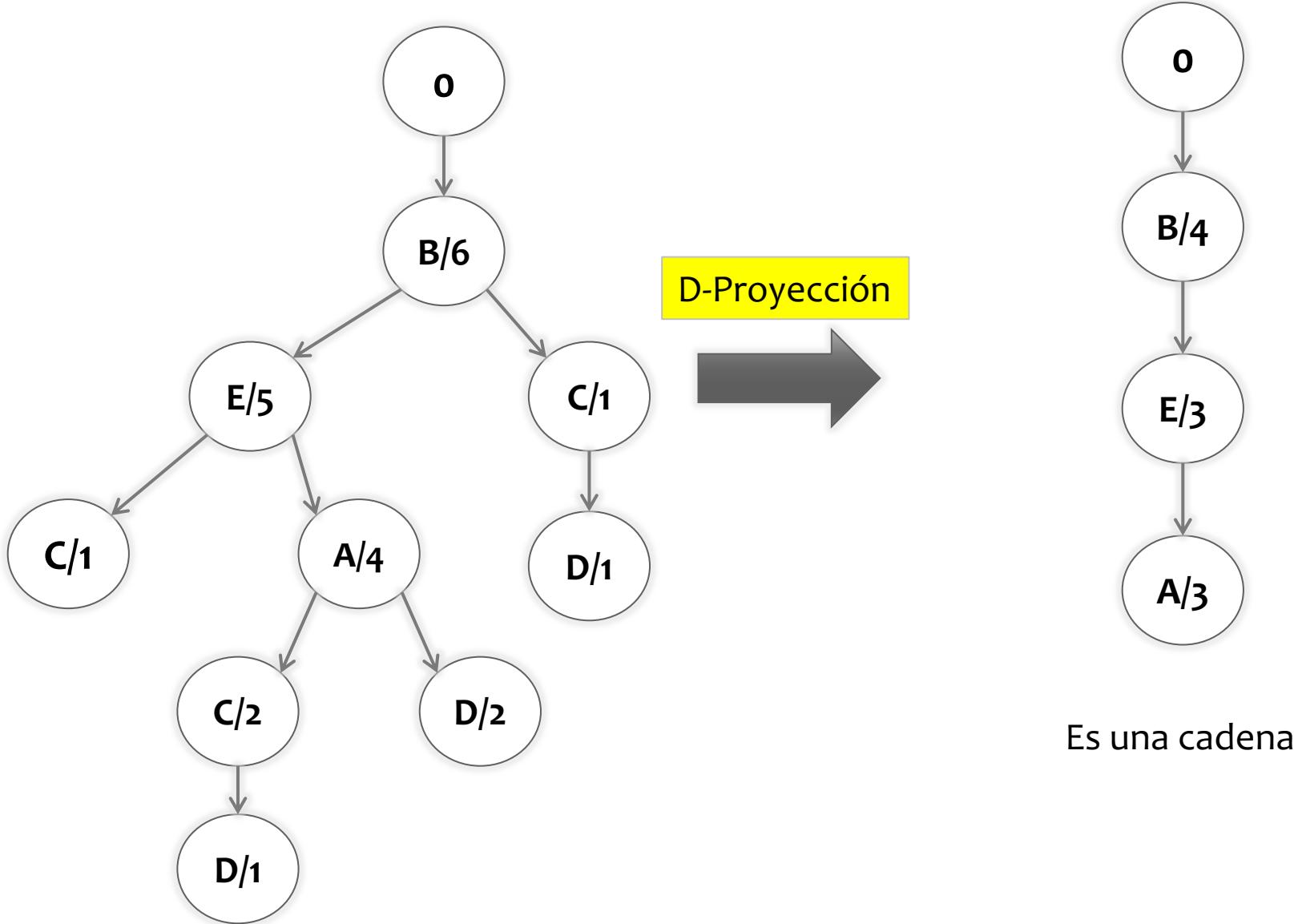
Frequent Items : D(4),



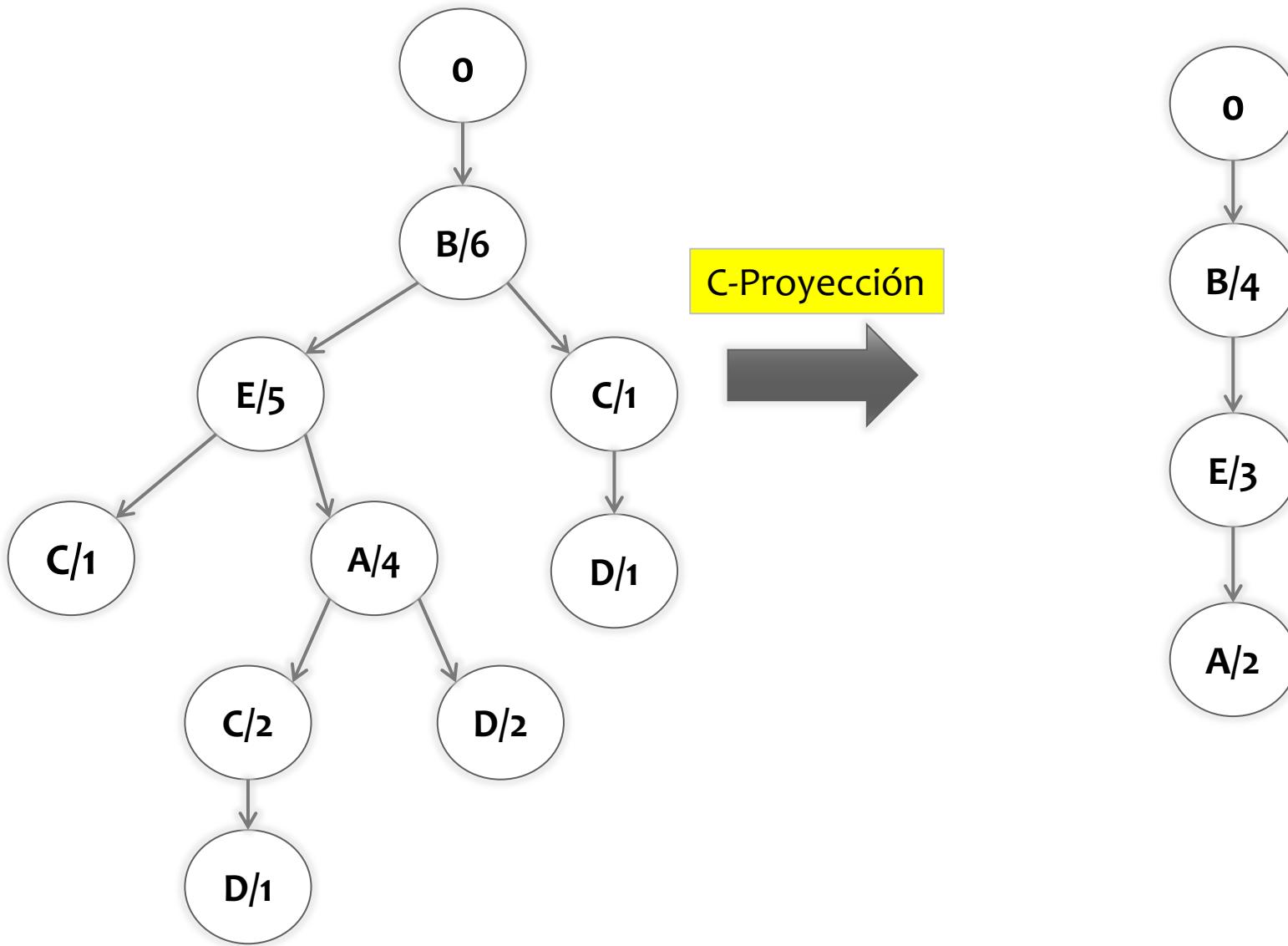
Frequent Items : D(4),



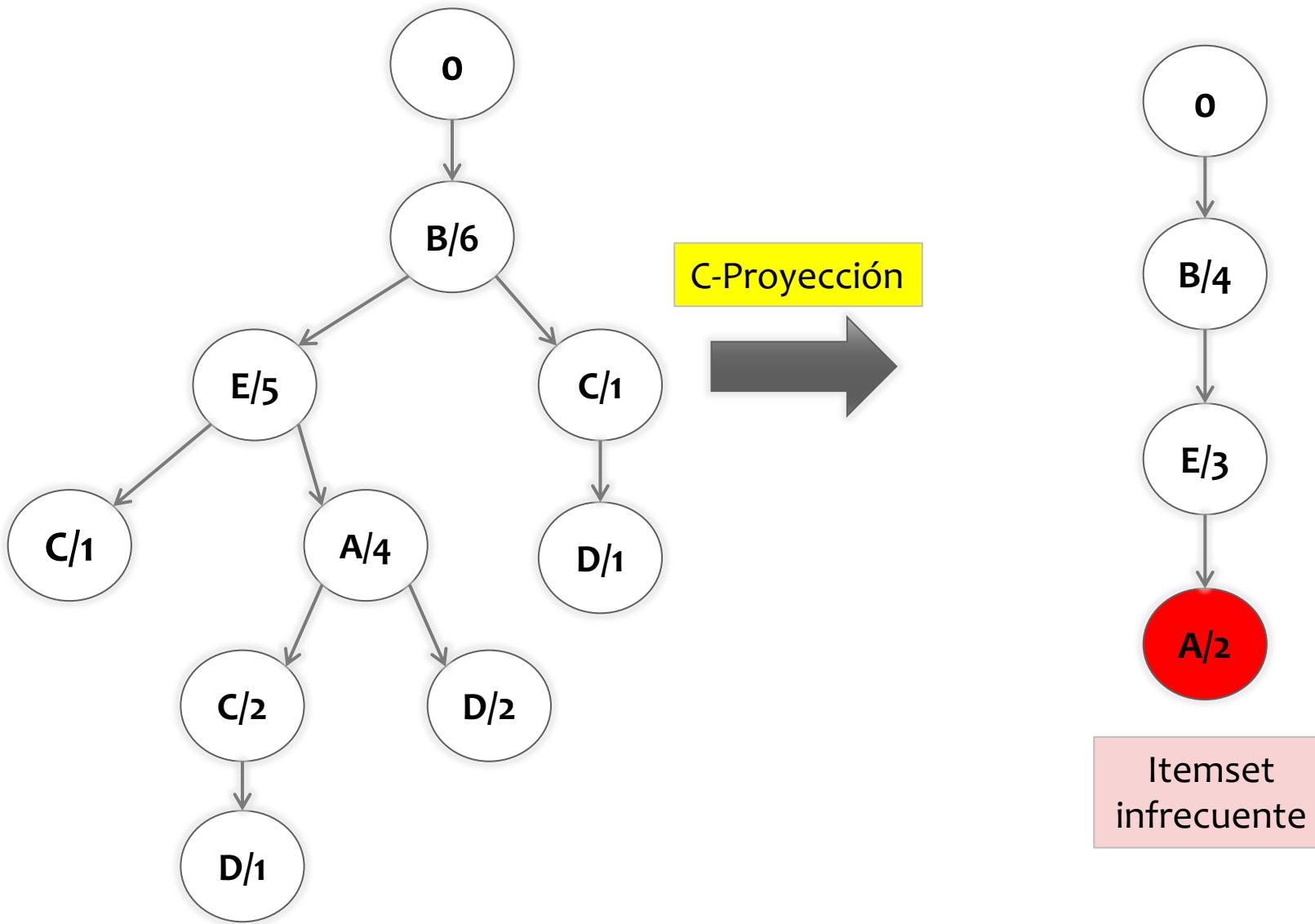
Frequent Items : D(4), DB(4), DE (3), DA(3),DBE(3),DBA(3), DEA(3),DBEA(3)



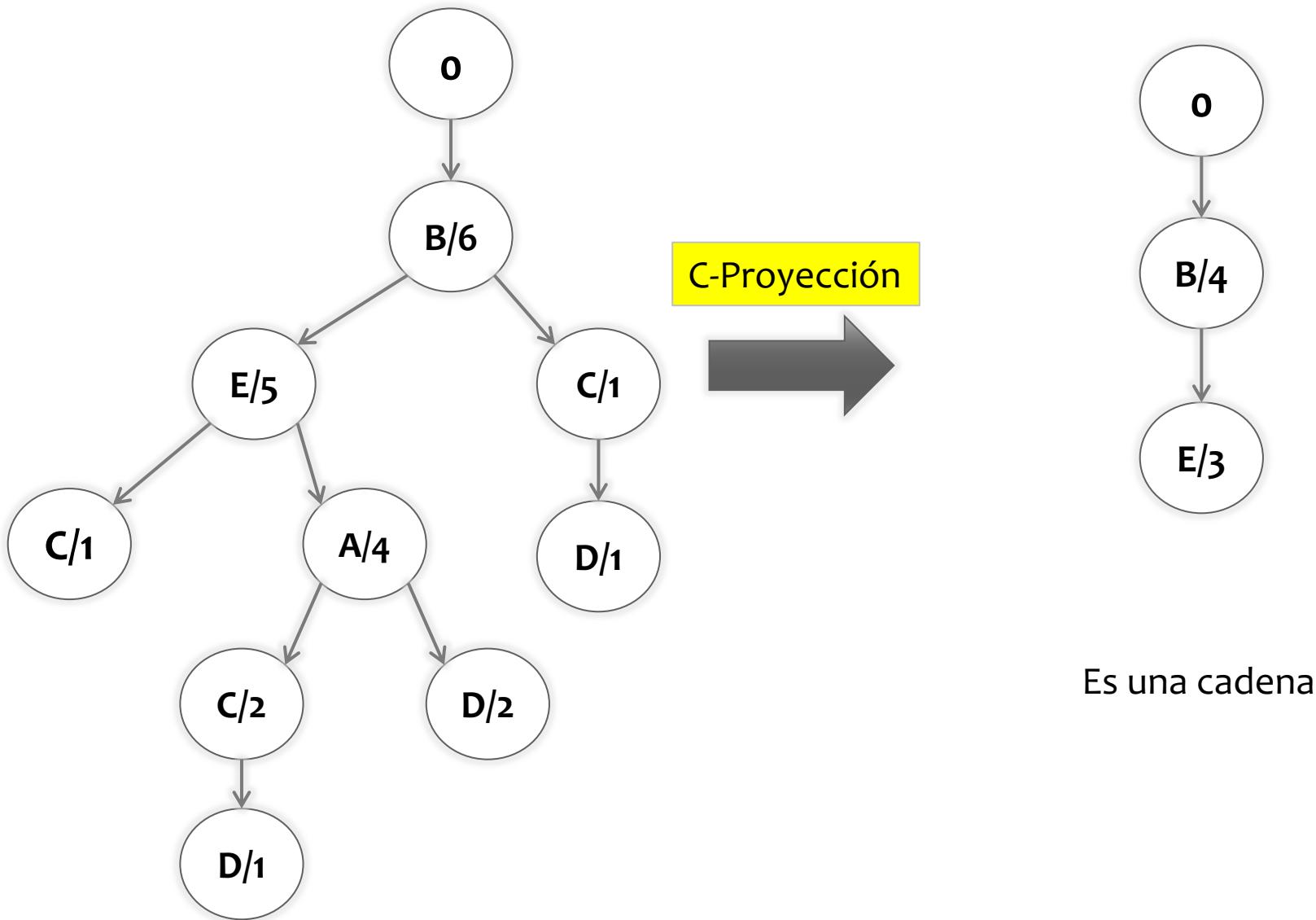
Frequent Items : D(4), DB(4), DE (3), DA(3),DBE(3),DBA(3), DEA(3),DBEA(3)  
C(4)



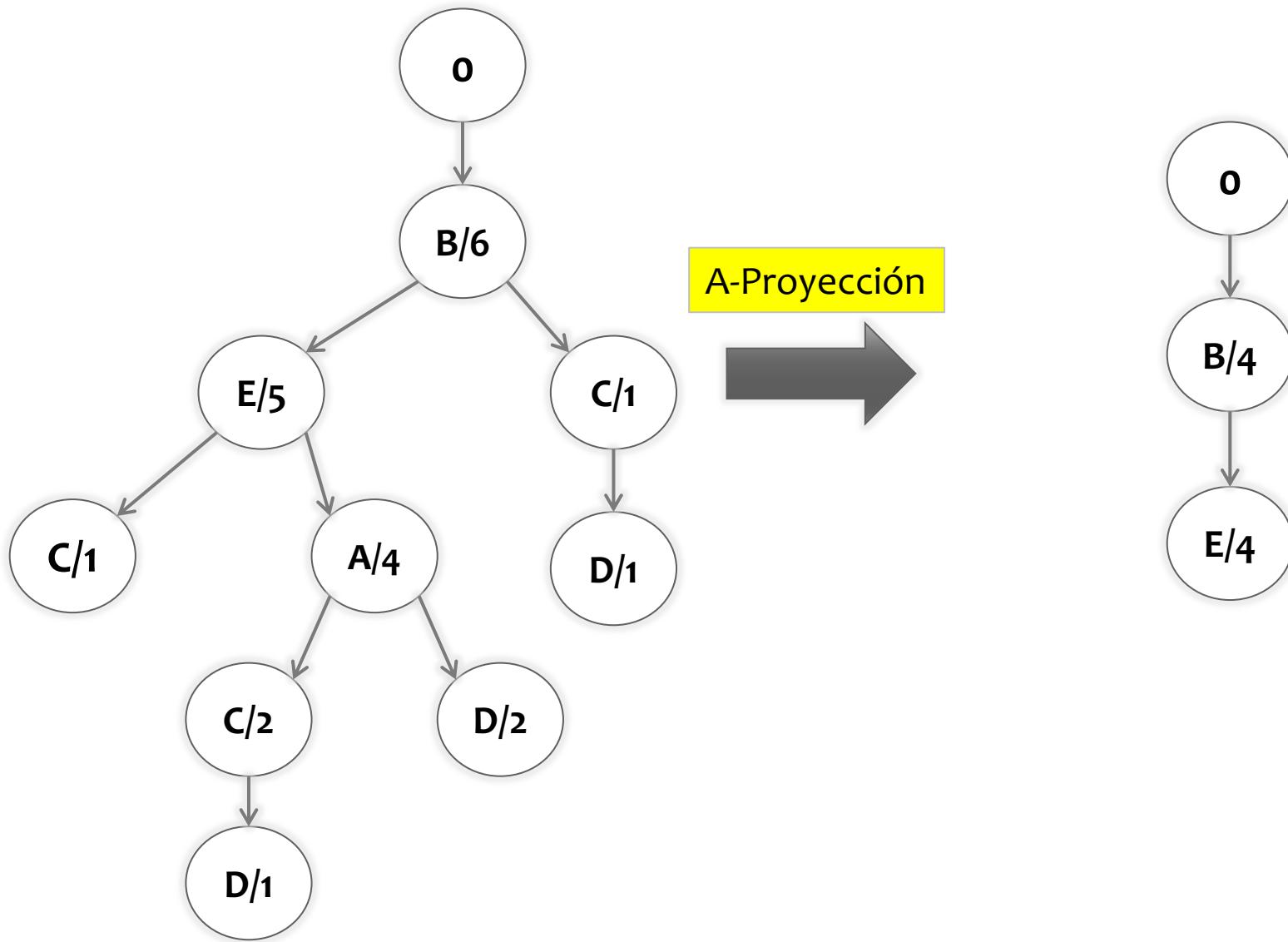
Frequent Items : D(4), DB(4), DE (3), DA(3),DBE(3),DBA(3), DEA(3),DBEA(3)  
C(4)



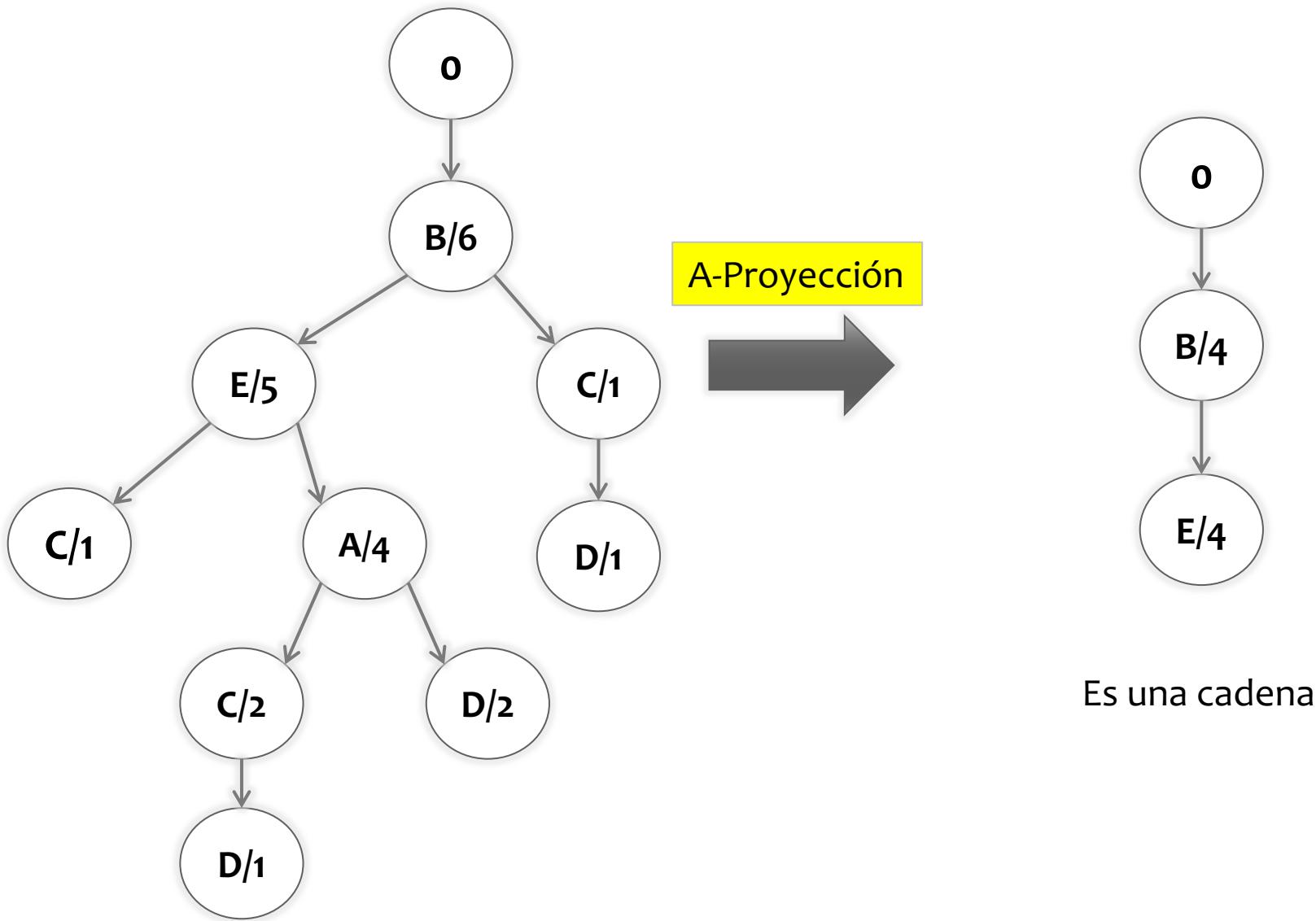
Frequent Items : D(4), DB(4), DE (3), DA(3),DBE(3),DBA(3), DEA(3),DBEA(3)  
C(4), CB(4), CE(3),CBE(3)



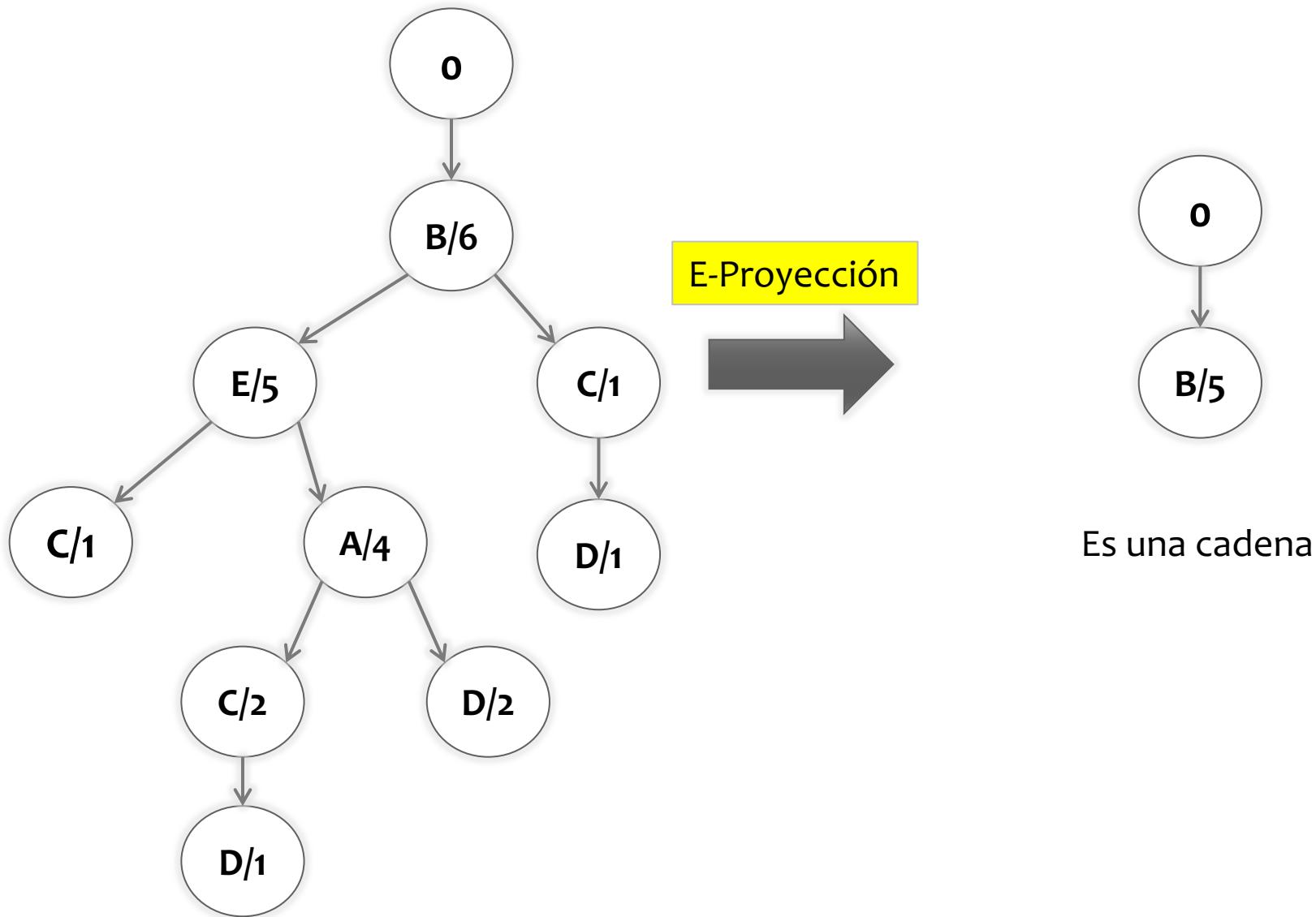
Frequent Items : D(4), DB(4), DE (3), DA(3),DBE(3),DBA(3), DEA(3),DBEA(3)  
C(4), CB(4), CE(3),CBE(3), A(4),



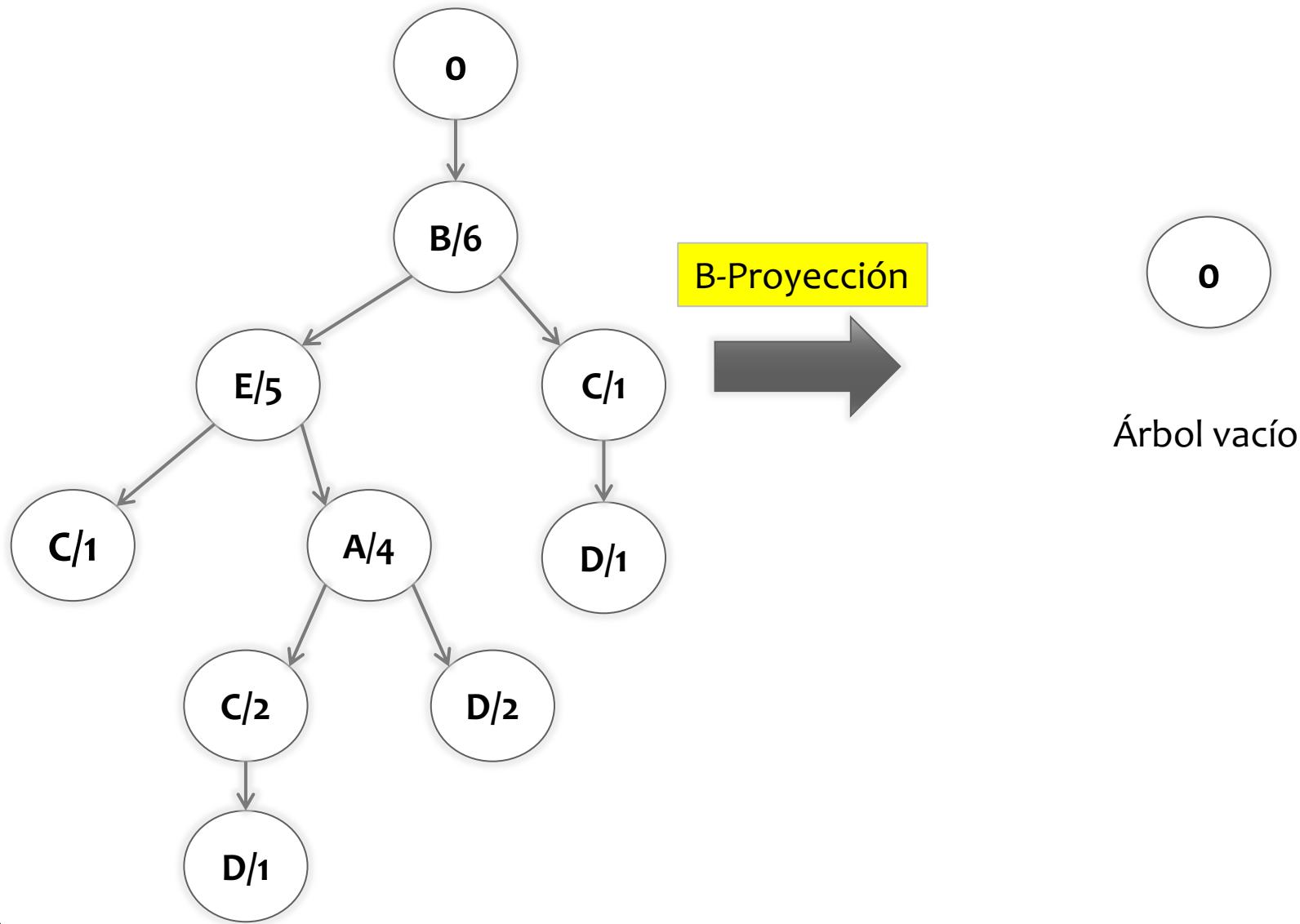
Frequent Items : D(4), DB(4), DE (3), DA(3),DBE(3),DBA(3), DEA(3),DBEA(3)  
C(4), CB(4), CE(3),CBE(3), A(4), AB(4), AE(4), ABE(4)



Frequent Items : D(4), DB(4), DE (3), DA(3),DBE(3),DBA(3), DEA(3),DBEA(3)  
C(4), CB(4), CE(3),CBE(3), A(4), AB(4), AE(4), ABE(4), **E(5)**, **EB(5)**



Frequent Items : D(4), DB(4), DE (3), DA(3),DBE(3),DBA(3), DEA(3),DBEA(3)  
C(4), CB(4), CE(3),CBE(3), A(4), AB(4), AE(4), ABE(4), E(5), EB(5), **B(6)**



# Índice

- Concepto de regla de asociación
- Algoritmo Apriori
- Variaciones del algoritmo A priori
- Algoritmo FP-Growth
- Generación de reglas
- Evaluación de reglas
- Reglas difusas
- Otros aspectos

# Generación de reglas

Dado un itemset frecuente  $L$ , se trata de encontrar todos los subconjuntos no vacíos  $f \subset L$  tales que  $f \rightarrow L - f$  satisfaga el umbral de confianza mínima (MinConf)

## Ejemplo

A partir del itemset frecuente  $\{A, B, C, D\}$ , se generan las siguientes reglas candidatas:

- $A \rightarrow BCD$ ,  $B \rightarrow ACD$ ,  $C \rightarrow ABD$ ,  $D \rightarrow ABC$ ,
- $AB \rightarrow CD$ ,  $AC \rightarrow BD$ ,  $AD \rightarrow BC$ ,  $BC \rightarrow AD$ ,  $BD \rightarrow AC$ ,  $CD \rightarrow AB$ ,
- $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ACD \rightarrow B$ ,  $BCD \rightarrow A$

# Generación de reglas

Dado un itemset frecuente  $L$ , se trata de encontrar todos los subconjuntos no vacíos  $f \subset L$  tales que  $f \rightarrow L - f$  satisfaga el umbral de confianza mínima ( $\text{MinConf}$ )

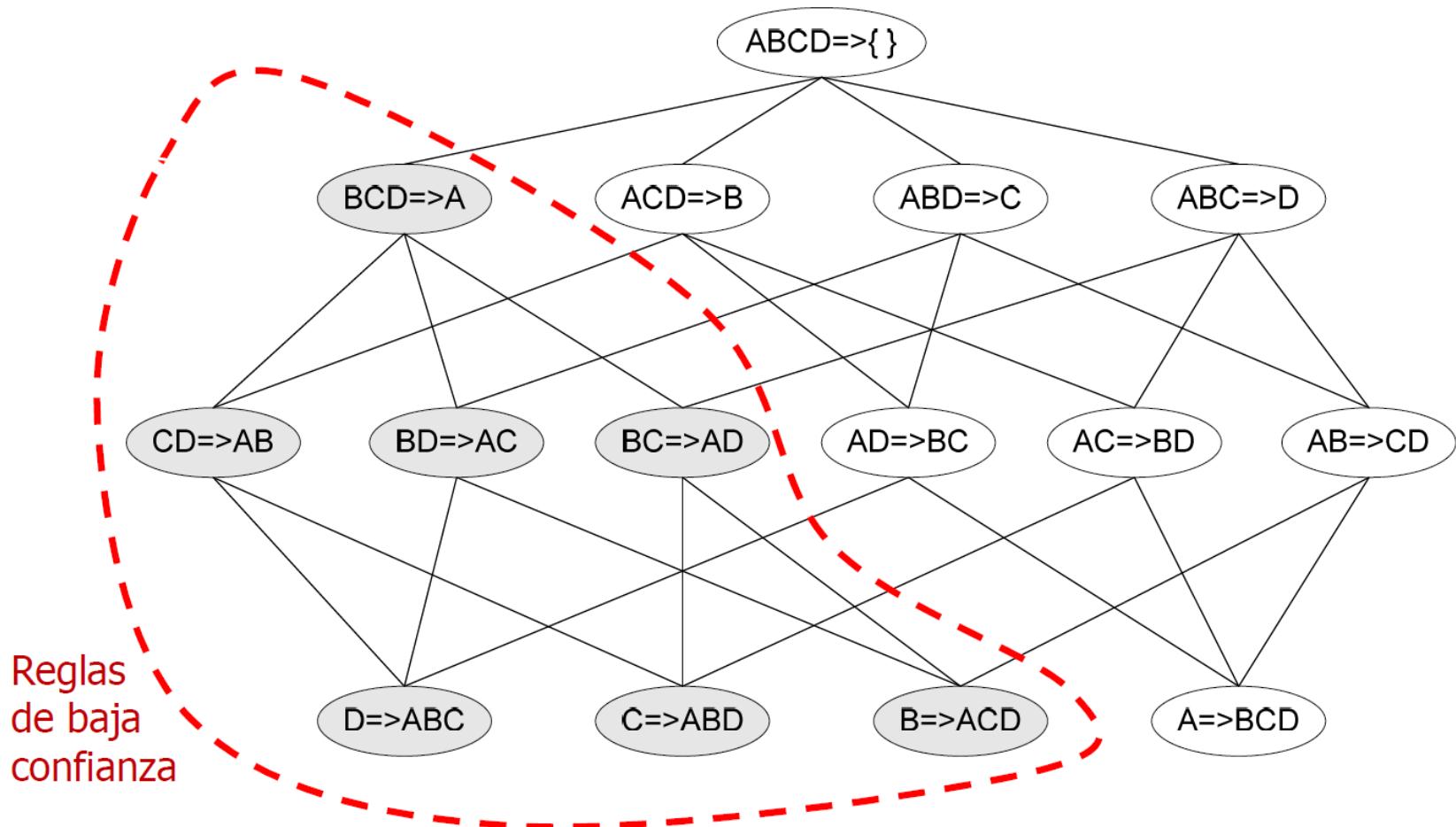
Calcular todas las reglas posibles y quedarse con las que tienen confianza mayor que  $\text{MinConf}$ , no es eficiente

Si  $|L| = k$ , entonces hay  $2^{k-2}$  reglas de asociación candidatas (ignorando  $L \rightarrow \emptyset$  y  $\emptyset \rightarrow L$ ). Eficiencia exponencial!

# Generación de reglas

- ❑ ¿Es la **confianza antimonótona** como el soporte?
  - NO: La confianza de  $ABC \rightarrow D$  puede ser mayor o menor que la confianza de  $AB \rightarrow D$
- ❑ Pero la confianza de las reglas generadas de un mismo itemset tienen una propiedad antimonótona
  - Para  $L = \{A, B, C, D\}$
  - $\text{Conf}(ABC \rightarrow D) \geq \text{Conf}(AB \rightarrow CD) \geq \text{Conf}(A \rightarrow BCD)$
- ❑ La confianza es antimonótona con respecto al número de items en la parte derecha de la regla.

# Generación de reglas



# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

$$\text{Supp}(A)=15/20=0.75$$

$$\text{Supp}(B)=16/20=0.8$$

$$\text{Supp}(C)=9/20=0.45$$

$$\text{Supp}(D)=6/20=0.3$$

$$\text{Supp}(E)=13/20=0.65$$

$$\text{Supp}(F)=6/20=0.3$$

# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

$$\begin{aligned} \text{Supp}(A) &= 15/20 = 0.75 \\ \text{Supp}(B) &= 16/20 = 0.8 \\ \text{Supp}(C) &= 9/20 = 0.45 \\ \text{Supp}(D) &= 6/20 = 0.3 \\ \text{Supp}(E) &= 13/20 = 0.65 \\ \text{Supp}(F) &= 6/20 = 0.3 \end{aligned}$$

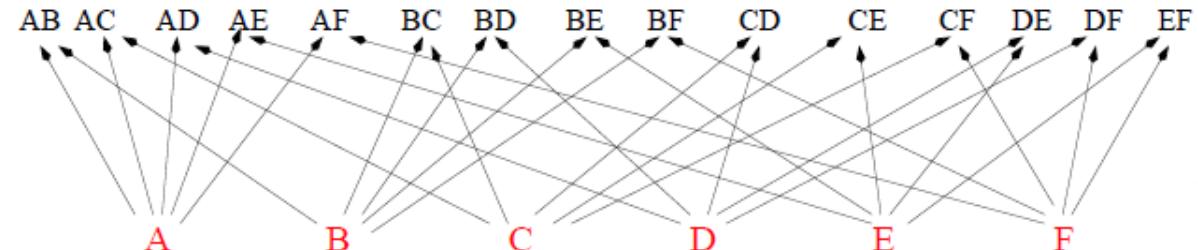
Todos superan  
MinSupp

# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

$$\begin{aligned}
 \text{Supp}(A) &= 15/20 = 0.75 \\
 \text{Supp}(B) &= 16/20 = 0.8 \\
 \text{Supp}(C) &= 9/20 = 0.45 \\
 \text{Supp}(D) &= 6/20 = 0.3 \\
 \text{Supp}(E) &= 13/20 = 0.65 \\
 \text{Supp}(F) &= 6/20 = 0.3
 \end{aligned}$$

Todos superan  
MinSupp

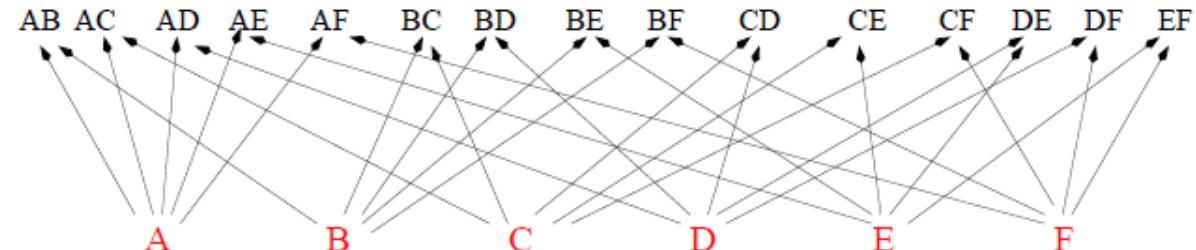


# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

$\text{Supp}(A,B) = 11/20$   
 $\text{Supp}(B,C) = 6/20$   
 $\text{Supp}(C,D) = 4/20$   
 $\text{Supp}(D,E) = 2/20$   
 $\text{Supp}(A,C) = 6/20$   
 $\text{Supp}(B,D) = 4/20$   
 $\text{Supp}(C,E) = 4/20$

$\text{Supp}(D,F) = 2/20$   
 $\text{Supp}(A,D) = 4/20$   
 $\text{Supp}(B,E) = 11/20$   
 $\text{Supp}(C,F) = 5/20$   
 $\text{Supp}(E,F) = 4/20$   
 $\text{Supp}(A,E) = 10/20$   
 $\text{Supp}(B,F) = 4/20$   
 $\text{Supp}(A,F) = 4/20$



# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

**Supp(A,B)= 11/20**

**Supp(B,C)= 6/20**

Supp(C,D)= 4/20

Supp(D,E)= 2/20

**Supp(A,C)= 6/20**

Supp(B,D)= 4/20

Supp(C,E)= 4/20

Supp(D,F)= 2/20

Supp(A,D)= 4/20

**Supp(B,E)= 11/20**

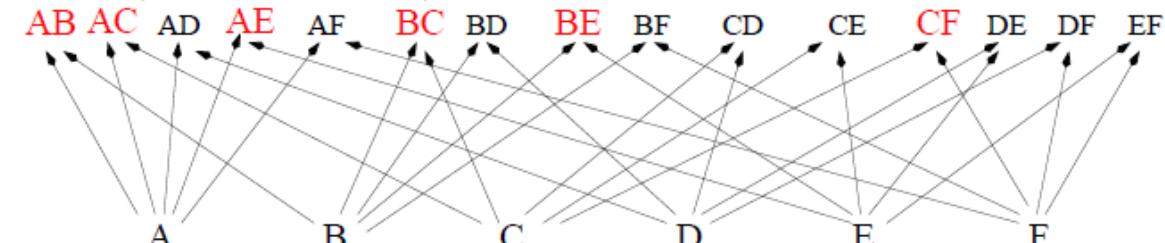
**Supp(C,F)= 5/20**

Supp(E,F)= 4/20

**Supp(A,E)= 10/20**

Supp(B,F)= 4/20

Supp(A,F)= 4/20

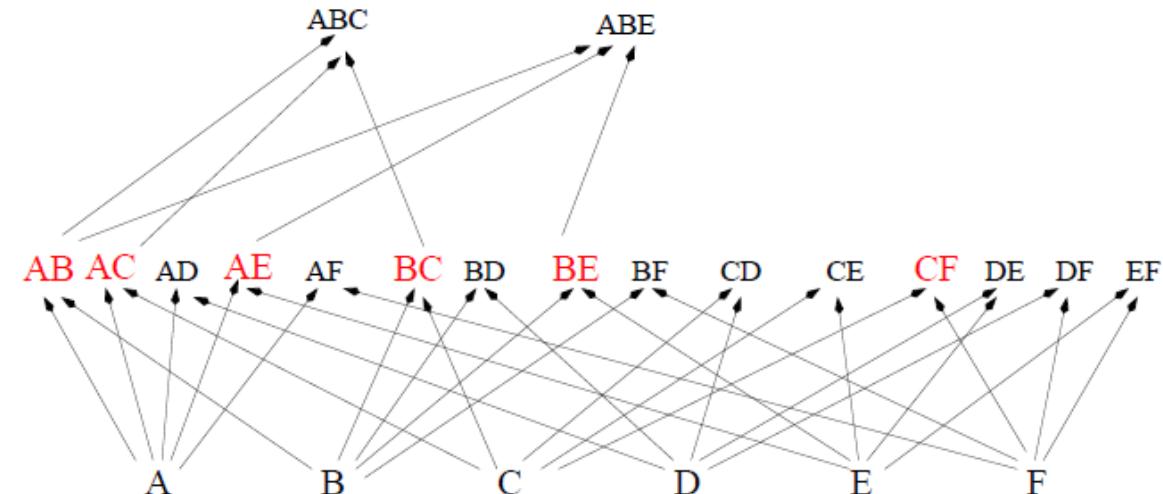


# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

Itemset frecuentes: AB, AC, AE, BC, BE, CF

$$\text{Supp}(A, B, C) = 3/20$$
$$\text{Supp}(A, B, E) = 8/20$$

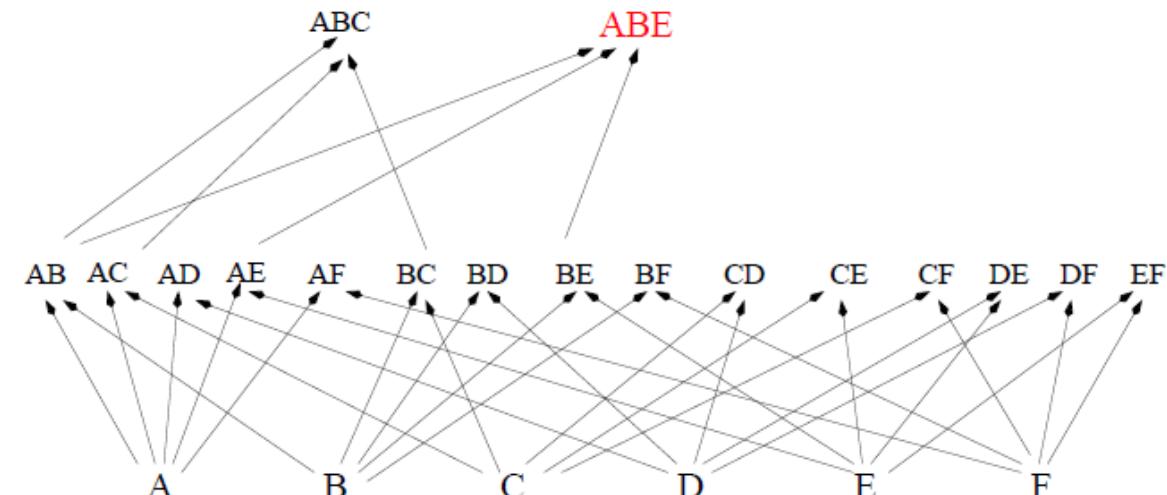


# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

Itemset frecuentes: AB, AC, AE, BC, BE, CF, ABE

$$\text{Supp}(A, B, C) = 3/20$$
$$\text{Supp}(A, B, E) = 8/20$$

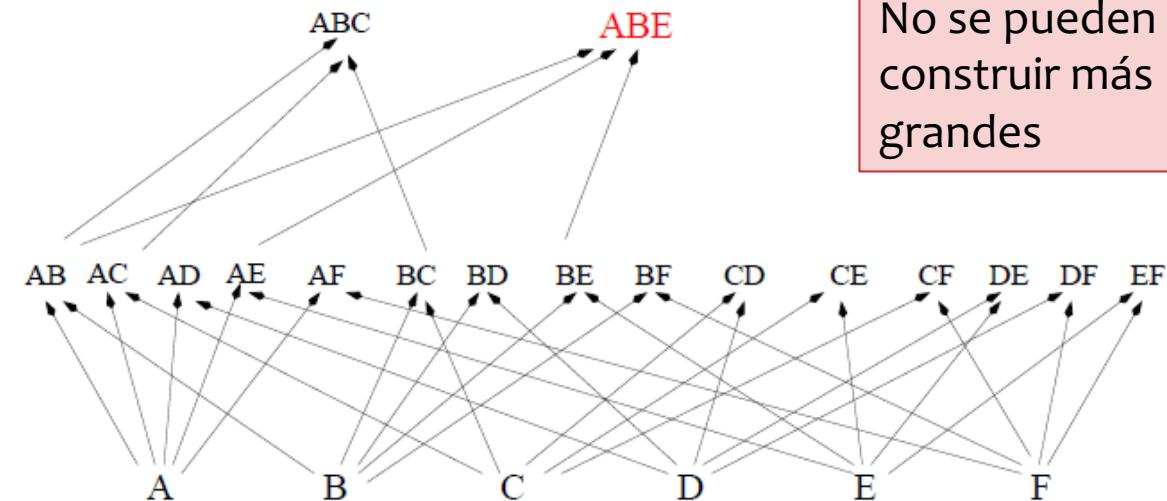


# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

Itemset frecuentes: AB, AC, AE, BC, BE, CF, ABE

$$\text{Supp}(A, B, C) = 3/20$$
$$\text{Supp}(A, B, E) = 8/20$$



# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

Itemset frecuentes: AB, AC, AE, BC, BE, CF, ABE

**Conf(A→B)=11/15**

Conf(B→A)=11/16

Conf(A→C)=6/15

Conf(C→A)=6/9

Conf(A→E)=10/15

**Conf(E→A)=10/13**

Conf(B→C)=6/16

Conf(C→B)=6/9

Conf(B→E)=11/16

**Conf(E→B)=11/13**

Conf(C→F)=5/9

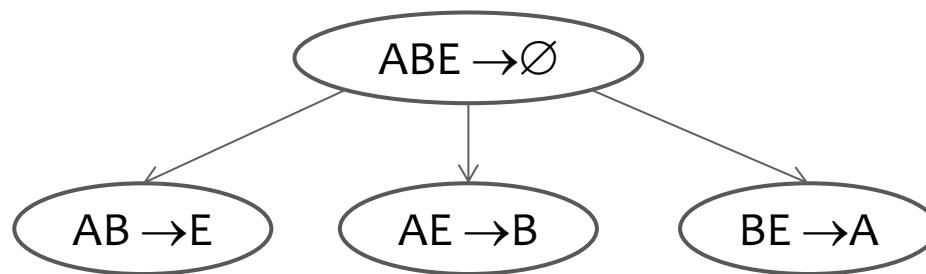
**Conf(F→C)=5/6**

# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

Itemset frecuentes: AB, AC, AE, BC, BE, CF, ABE

Reglas:  $A \rightarrow B$ ,  $E \rightarrow A$ ,  $E \rightarrow B$ ,  $F \rightarrow C$

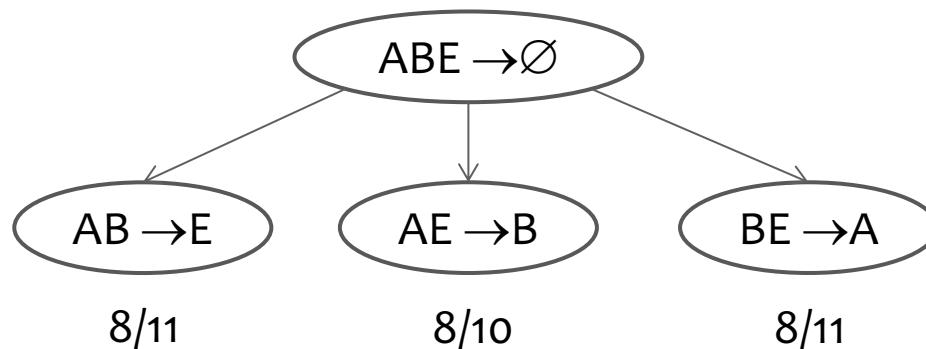


# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

Itemset frecuentes: AB, AC, AE, BC, BE, CF, ABE

Reglas:  $A \rightarrow B$ ,  $E \rightarrow A$ ,  $E \rightarrow B$ ,  $F \rightarrow C$

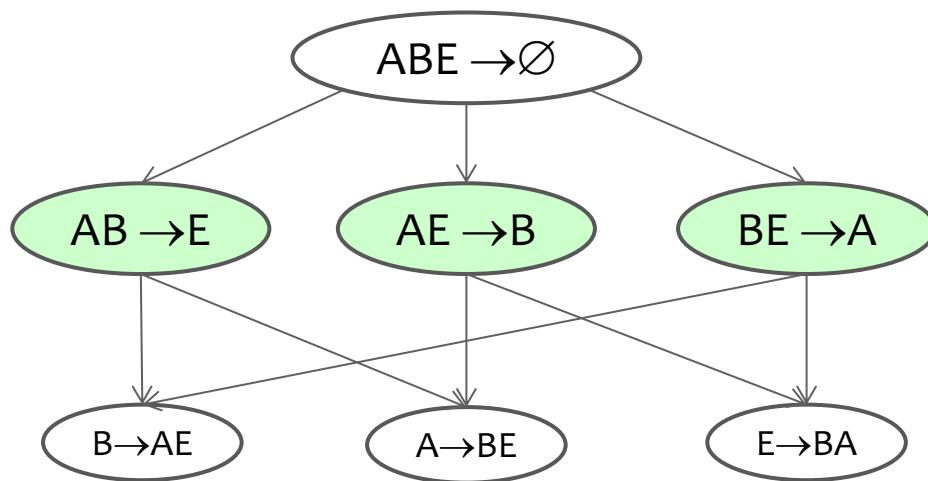


# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

Itemset frecuentes: AB, AC, AE, BC, BE, CF, ABE

Reglas: A→B, E→A, E→B, F→C

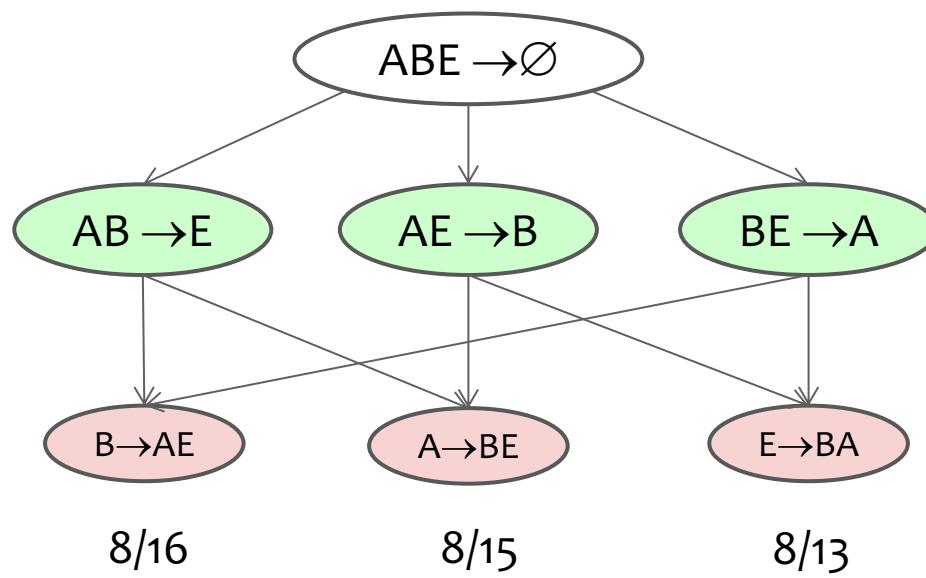


# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

Itemset frecuentes: AB, AC, AE, BC, BE, CF, ABE

Reglas: A→B, E→A, E→B, F→C



# Algoritmo Apriori (MinSupp=0.25, MinConf=0.7)

	A	B	C	D	E	F
1	1	1	0	0	1	0
2	1	0	1	0	1	1
3	0	1	1	1	0	1
4	1	1	1	0	0	0
5	0	1	0	0	1	1
6	1	0	1	1	0	0
7	1	1	0	0	1	0
8	1	1	0	0	0	0
9	1	0	1	1	0	1
10	1	1	1	0	1	1
11	1	1	0	0	1	0
12	1	1	0	1	1	0
13	1	1	1	1	1	1
14	0	1	1	0	0	0
15	1	1	0	1	0	0
16	1	1	0	0	1	0
17	0	1	0	0	1	0
18	0	1	1	0	1	0
19	1	0	0	0	1	0
20	1	1	0	0	1	0

**Itemset frecuentes:** AB, AC, AE, BC, BE, CF, ABE

**Reglas:** A → B, E → A, E → B, F → C, AB → E, AE → B, BE → A

# Filtrado

Se pueden generar un número de reglas excesivo: varios miles como mínimo.

## Soluciones

- Guiado por el usuario.
  - El usuario establece a priori las reglas que él considera interesantes y el sistema las compara con las obtenidas
  - El usuario visualiza las reglas y selecciona la parte de ellas que le interesa
- Sin ayuda del usuario.
  - Ordenar las reglas según un grado de interés calculado con procedimientos estadísticos
  - Usando medidas de bondad alternativas mucho más restrictivas, por lo que, obviamente, salen menos reglas
  - Con el cálculo de otros itemsets distintos de los frecuentes (cerrados, maximales, interesantes,...)

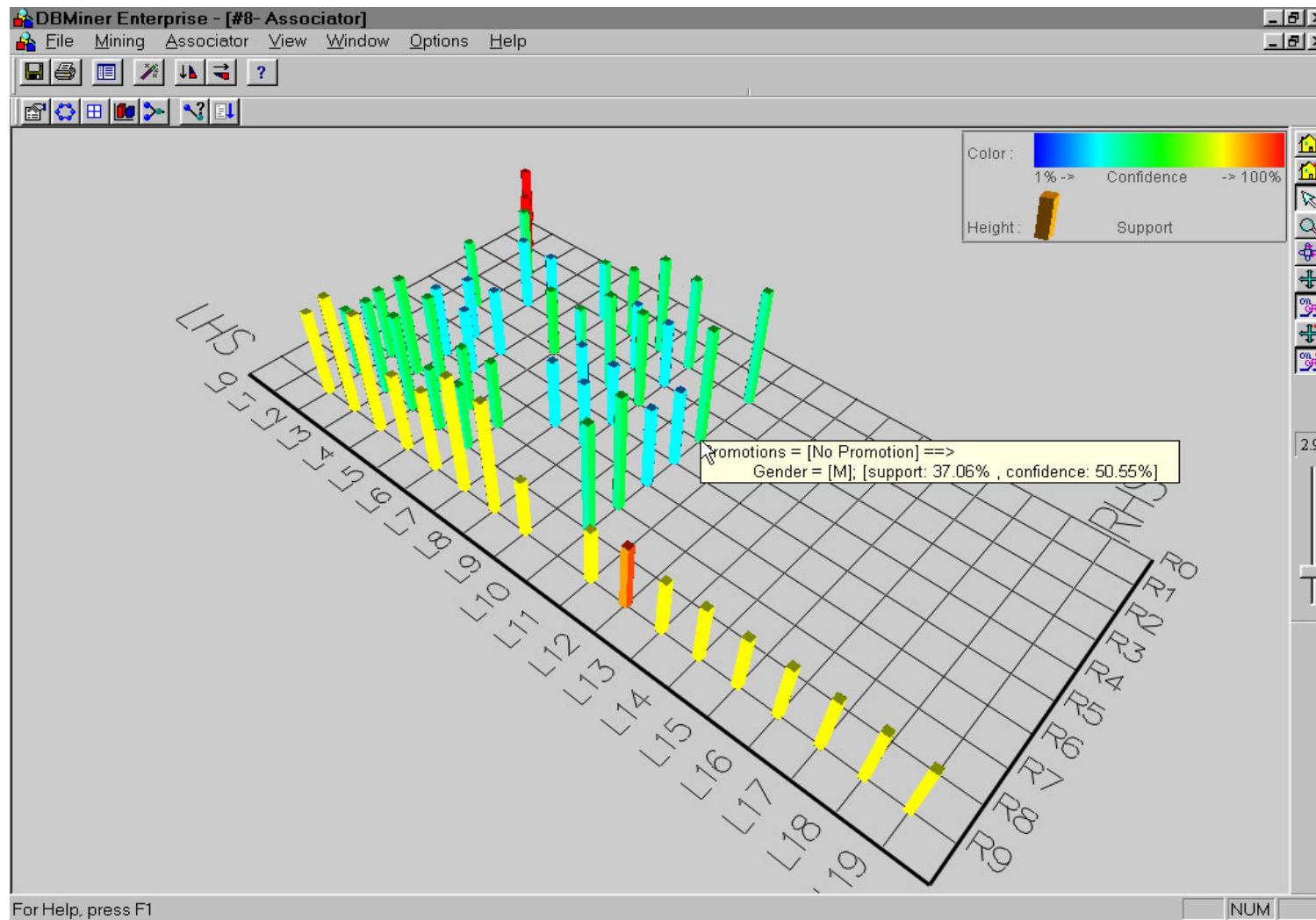
# Visualización de reglas

## Tablas

	Body	Implies	Head	Supp (%)	Conf (%)	F	G	H	I	J
1	$\text{cost}(x) = 0.00\text{--}1000.00'$	$\implies$	$\text{revenue}(x) = 0.00\text{--}500.00'$	28.45	40.4					
2	$\text{cost}(x) = 0.00\text{--}1000.00'$	$\implies$	$\text{revenue}(x) = 500.00\text{--}1000.00'$	20.46	29.05					
3	$\text{cost}(x) = 0.00\text{--}1000.00'$	$\implies$	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	59.17	84.04					
4	$\text{cost}(x) = 0.00\text{--}1000.00'$	$\implies$	$\text{revenue}(x) = 1000.00\text{--}1500.00'$	10.45	14.84					
5	$\text{cost}(x) = 0.00\text{--}1000.00'$	$\implies$	$\text{region}(x) = \text{'United States'}$	22.56	32.04					
6	$\text{cost}(x) = 1000.00\text{--}2000.00'$	$\implies$	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	12.91	69.34					
7	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	$\implies$	$\text{revenue}(x) = 0.00\text{--}500.00'$	28.45	34.54					
8	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	$\implies$	$\text{cost}(x) = 1000.00\text{--}2000.00'$	12.91	15.67					
9	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	$\implies$	$\text{region}(x) = \text{'United States'}$	25.9	31.45					
10	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	$\implies$	$\text{cost}(x) = 0.00\text{--}1000.00'$	59.17	71.86					
11	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	$\implies$	$\text{product\_line}(x) = \text{'Tents'}$	13.52	16.42					
12	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	$\implies$	$\text{revenue}(x) = 500.00\text{--}1000.00'$	19.67	23.88					
13	$\text{product\_line}(x) = \text{'Tents'}$	$\implies$	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	13.52	98.72					
14	$\text{region}(x) = \text{'United States'}$	$\implies$	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	25.9	81.94					
15	$\text{region}(x) = \text{'United States'}$	$\implies$	$\text{cost}(x) = 0.00\text{--}1000.00'$	22.56	71.39					
16	$\text{revenue}(x) = 0.00\text{--}500.00'$	$\implies$	$\text{cost}(x) = 0.00\text{--}1000.00'$	28.45	100					
17	$\text{revenue}(x) = 0.00\text{--}500.00'$	$\implies$	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	28.45	100					
18	$\text{revenue}(x) = 1000.00\text{--}1500.00'$	$\implies$	$\text{cost}(x) = 0.00\text{--}1000.00'$	10.45	96.75					
19	$\text{revenue}(x) = 500.00\text{--}1000.00'$	$\implies$	$\text{cost}(x) = 0.00\text{--}1000.00'$	20.46	100					
20	$\text{revenue}(x) = 500.00\text{--}1000.00'$	$\implies$	$\text{order\_qty}(x) = 0.00\text{--}100.00'$	19.67	96.14					
21										
22										
23	$\text{cost}(x) = 0.00\text{--}1000.00'$	$\implies$	$\text{revenue}(x) = 0.00\text{--}500.00' \text{ AND } \text{order\_qty}(x) = 0.00\text{--}100.00'$	28.45	40.4					
24	$\text{cost}(x) = 0.00\text{--}1000.00'$	$\implies$	$\text{revenue}(x) = 0.00\text{--}500.00' \text{ AND } \text{order\_qty}(x) = 0.00\text{--}100.00'$	28.45	40.4					
25	$\text{cost}(x) = 0.00\text{--}1000.00'$	$\implies$	$\text{revenue}(x) = 500.00\text{--}1000.00' \text{ AND } \text{order\_qty}(x) = 0.00\text{--}100.00'$	19.67	27.93					
26	$\text{cost}(x) = 0.00\text{--}1000.00'$	$\implies$	$\text{revenue}(x) = 500.00\text{--}1000.00' \text{ AND } \text{order\_qty}(x) = 0.00\text{--}100.00'$	19.67	27.93					
27	$\text{cost}(x) = 0.00\text{--}1000.00' \text{ AND } \text{order\_qty}(x) = 0.00\text{--}100.00'$	$\implies$	$\text{revenue}(x) = 500.00\text{--}1000.00'$	19.67	33.23					

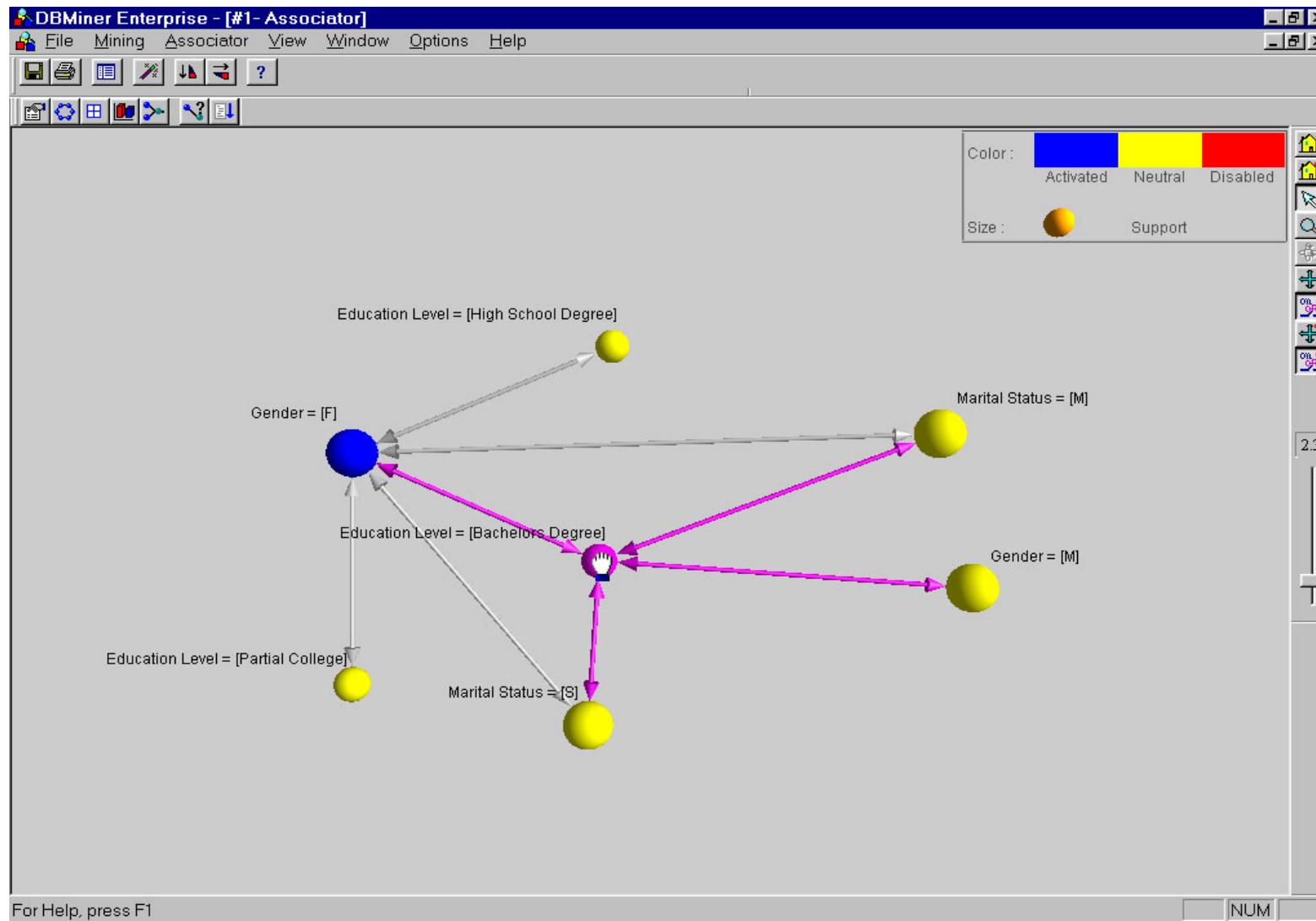
# Visualización de reglas

## Gráfico 2D



# Visualización de reglas

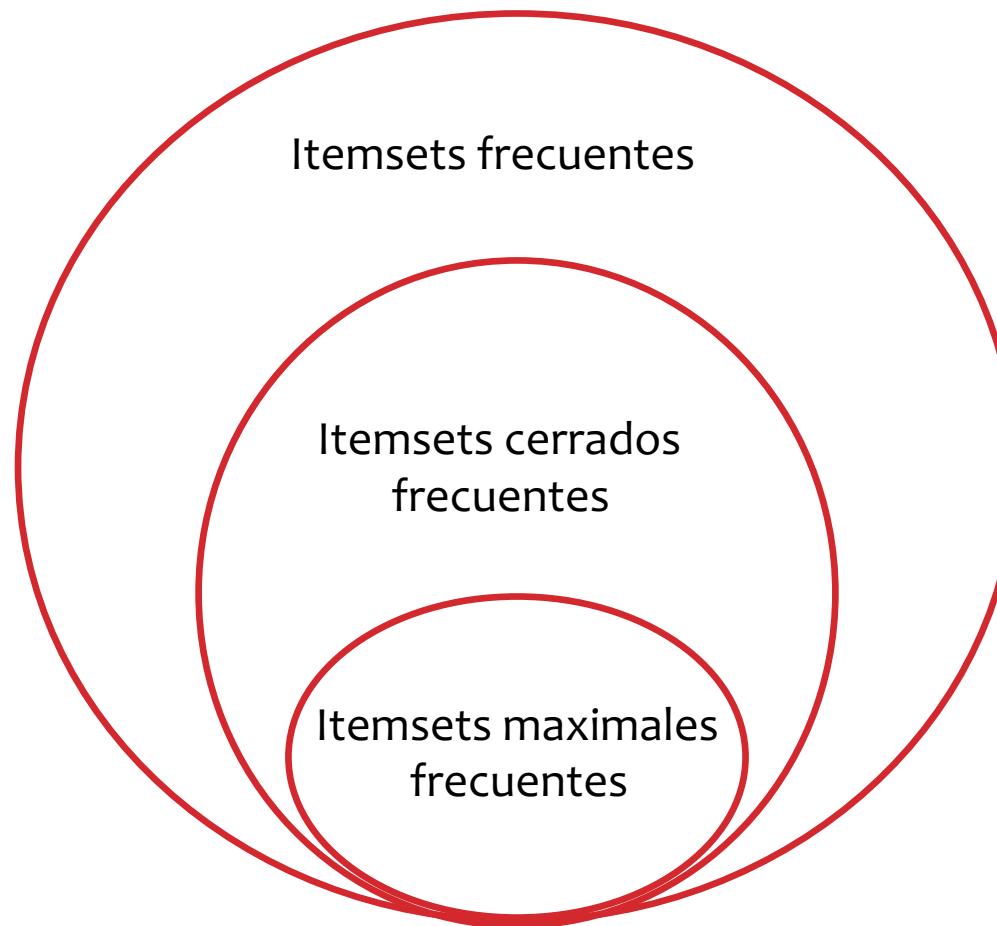
## Grafo



# Itemsets maximales y cerrados

- Un itemset es **cerrado** (closed) si no existe un superitemset suyo (es decir, que lo contenga) que tenga el mismo soporte
  - Se busca encontrar los itemsets cerrados y frecuentes
  - Enumerando los itemsets cerrados frecuentes tenemos la información completa de los itemsets frecuentes
- Un itemset es **maximal** y frecuente (max-itemset) si es frecuente y no existe un superitemset que también sea frecuente
  - Se pueden calcular los itemsets frecuentes a partir de los maximales, pero no su soporte.

# Itemsets maximales y cerrados



# Atributos continuos

Los atributos continuos no son directamente utilizables en reglas de asociación. Es necesario **discretizar** previamente

$\text{Edad} \in [21, 35) \wedge \text{Salario} \in [40k, 60k) \rightarrow \text{Compra}$

$\text{Salario} \in [30k, 60k) \wedge \text{Compra} \rightarrow \text{Edad} \in [17, 24)$

Otra opción: Reglas de asociación difusas

# Índice

- Concepto de regla de asociación
- Algoritmo Apriori
- Variaciones del algoritmo A priori
- Algoritmo FP-Growth
- Generación de reglas
- **Evaluación de reglas**
- Reglas difusas
- Otros aspectos

# Evaluación de reglas

La confianza no es la mejor medida de interés posible para las reglas de asociación

	Café	No café
Té	15	5
No té	75	5

- La regla Té → Café tiene una confianza del 75%
- Pero no es un resultado lógico, comprar té disminuye la probabilidad de comprar café

Problema: Café es muy frecuente y cualquier antecedente produce una regla con alta confianza

# Evaluación de reglas

## Lift

Es una medida de la correlación entre antecedente y consecuente

$$\text{Lift}(A \rightarrow B) = \frac{P(A \cup B)}{P(A)P(B)} = \frac{\text{Conf}(A \rightarrow B)}{\text{Supp}(B)}$$

- Si es menor que 1, están negativamente correlacionados
- Si es mayor que 1, están positivamente correlacionados
- Si es 1, entonces son independientes

# Evaluación de reglas

## Ejemplo

	Café	No café
Té	15	5
No té	75	5

$$\text{Lift}(\text{Té} \rightarrow \text{Café}) = 0.75/0.9 = 0.833$$

- Están negativamente correlacionados
- Esto no podría deducirse de la confianza

Las reglas se evalúan con una terna (soporte,confianza,correlación)

# Evaluación de reglas

## Inconvenientes de lift

- Simetría

$$\text{Lift}(\text{Té} \rightarrow \text{Café}) = \text{Lift}(\text{Café} \rightarrow \text{Té})$$

- Es sensible a reglas con items poco frecuentes en bases de datos pequeñas

	Y	$\bar{Y}$	
X	10	0	10
$\bar{X}$	0	90	90
	10	90	100

	Y	$\bar{Y}$	
X	90	0	90
$\bar{X}$	0	10	10
	90	10	100

$$Lift = \frac{0.1}{(0.1)(0.1)} = 10$$

$$Lift = \frac{0.9}{(0.9)(0.9)} = 1.11$$

# Evaluación de reglas

Otras medidas (null-trasaction invariant)

□ All\_confidence

$$\frac{\text{Sup}(AB)}{\max\{\text{Supp}(A), \text{Supp}(B)\}} = \min\{P(A|B), P(B|A)\}$$

□ Max\_confidence

$$\max\{P(A|B), P(B|A)\}$$

□ Coseno

$$\sqrt{P(A|B)P(B|A)}$$

# Evaluación de reglas

Otras medidas (null-trasaction invariant)

- Imbalance ratio

$$IR(A, B) = \frac{|\text{Supp}(A) - \text{Supp}(B)|}{\text{Supp}(A) + \text{Supp}(B) - \text{Supp}(AB)}$$

- Kulczynski

$$\frac{1}{2} (P(A|B) + P(B|A))$$

# Evaluación de reglas

Más medidas...

#	Measure	Formula
1	$\phi$ -coefficient	$\frac{P(A,B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's ( $\lambda$ )	$\frac{\sum_j \max_k P(A_j, B_k) + \sum_k \max_j P(A_j, B_k) - \max_j P(A_j) - \max_k P(B_k)}{2 - \max_j P(A_j) - \max_k P(B_k)}$
3	Odds ratio ( $\alpha$ )	$\frac{P(A,B)P(\bar{A},\bar{B})}{P(\bar{A},\bar{B})P(\bar{A},B)}$
4	Yule's $Q$	$\frac{P(A,B)P(\bar{A}\bar{B}) - P(A,\bar{B})P(\bar{A},B)}{P(A,B)P(\bar{A}\bar{B}) + P(A,\bar{B})P(\bar{A},B)} = \frac{\alpha - 1}{\alpha + 1}$
5	Yule's $Y$	$\frac{\sqrt{P(A,B)P(\bar{A}\bar{B})} - \sqrt{P(A,\bar{B})P(\bar{A},B)}}{\sqrt{P(A,B)P(\bar{A}\bar{B})} + \sqrt{P(A,\bar{B})P(\bar{A},B)}} = \frac{\sqrt{\alpha} - 1}{\sqrt{\alpha} + 1}$
6	Kappa ( $\kappa$ )	$\frac{P(A,B) + P(\bar{A},\bar{B}) - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}$ $\sum_i \sum_j P(A_i, B_j) \log \frac{P(A_i, B_j)}{P(A_i)P(B_j)}$
7	Mutual Information ( $M$ )	$\min(-\sum_i P(A_i) \log P(A_i), -\sum_j P(B_j) \log P(B_j))$
8	J-Measure ( $J$ )	$\max \left( P(A,B) \log \left( \frac{P(B A)}{P(B)} \right) + P(\bar{A}\bar{B}) \log \left( \frac{P(\bar{B} A)}{P(\bar{B})} \right), \right.$ $\left. P(A,B) \log \left( \frac{P(A B)}{P(A)} \right) + P(\bar{A}\bar{B}) \log \left( \frac{P(\bar{A} B)}{P(\bar{A})} \right) \right)$
9	Gini index ( $G$ )	$\max \left( P(A)[P(B A)^2 + P(\bar{B} A)^2] + P(\bar{A})[P(B \bar{A})^2 + P(\bar{B} \bar{A})^2] \right.$ $\left. - P(B)^2 - P(\bar{B})^2, \right.$ $\left. P(B)[P(A B)^2 + P(\bar{A} B)^2] + P(\bar{B})[P(A \bar{B})^2 + P(\bar{A} \bar{B})^2] \right.$ $\left. - P(A)^2 - P(\bar{A})^2 \right)$
10	Support ( $s$ )	$P(A,B)$
11	Confidence ( $c$ )	$\max(P(B A), P(A B))$
12	Laplace ( $L$ )	$\max \left( \frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2} \right)$
13	Conviction ( $V$ )	$\max \left( \frac{P(A)P(\bar{B})}{P(A\bar{B})}, \frac{P(B)P(\bar{A})}{P(B\bar{A})} \right)$
14	Interest ( $I$ )	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine ( $IS$ )	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's ( $PS$ )	$P(A,B) - P(A)P(B)$
17	Certainty factor ( $F$ )	$\max \left( \frac{P(B A)-P(B)}{1-P(B)}, \frac{P(A B)-P(A)}{1-P(A)} \right)$
18	Added Value ( $AV$ )	$\max(P(B A) - P(B), P(A B) - P(A))$
19	Collective strength ( $S$ )	$\frac{P(A,B)+P(\bar{A}\bar{B})}{P(A)P(B)+P(\bar{A})P(\bar{B})} \times \frac{1-P(A)P(B)-P(\bar{A})P(\bar{B})}{1-P(A,B)-P(\bar{A}\bar{B})}$
20	Jaccard ( $\zeta$ )	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen ( $K$ )	$\sqrt{P(A,B) \max(P(B A) - P(B), P(A B) - P(A))}$

# Índice

- Concepto de regla de asociación
- Algoritmo Apriori
- Variaciones del algoritmo A priori
- Algoritmo FP-Growth
- Generación de reglas
- Evaluación de reglas
- Reglas difusas
- Otros aspecto

# Reglas difusas

Los algoritmos para descubrir Reglas de Asociación en Bases de Datos Relacionales se han aplicado definiendo:

- Items como pares  $\langle$ atributo, valor $\rangle$
- Transacciones como tuplas

Una regla de asociación  $A \Rightarrow C$  establece que cualquier tupla que contenga los valores de atributo de A también contiene los valores de atributo de C.

# Reglas difusas

Los primeros trabajos solo consideraron asociaciones entre atributos categóricos

No obstante el problema de obtener Reglas de Asociación con atributos cuantitativos surgió muy poco después:

## Reglas de asociación cuantitativas

- Con valores numéricos, la granularidad más fina
- Pero no podemos calcular ítem frecuentes (para cada valor concreto, atributo=valor, posiblemente no sea frecuente)
- El coste computacional es demasiado elevado

# Reglas difusas

**Solución posible:** dividir el dominio de cada atributo cuantitativo en intervalos, y considerar el conjunto de estos como nuevo dominio (granularidad más basta).

Inconvenientes:

- Puede ser difícil encontrar semántica para los intervalos
- La importancia y precisión de las reglas puede ser muy sensible a pequeñas variaciones de las fronteras de los intervalos (**problema de la frontera**)

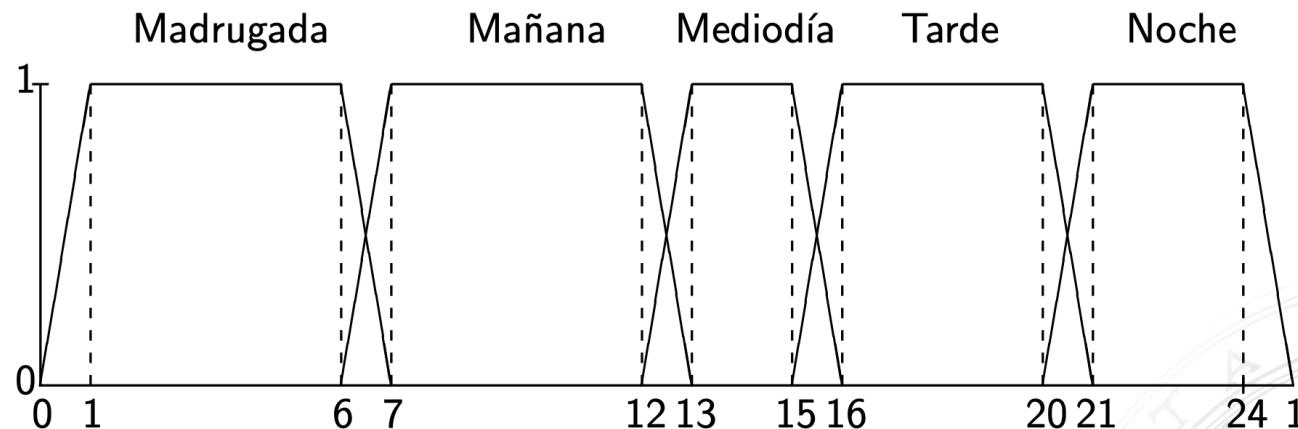
# Reglas difusas

Surge un uso natural de los **conjuntos difusos**. Definir un conjunto significativo de etiquetas lingüísticas (subconjuntos difusos) sobre el dominio de cada atributo cuantitativo, y usarlas como un nuevo dominio.

## Ejemplos

- Edad= {Bebe, Niño, Joven, Maduro, Anciano}
- Salario ={ Alto, Medio, Bajo }

Se suavizan  
las fronteras!



# Reglas difusa

Una transacción difusa es un subconjunto difuso de I

$$\tau : I \rightarrow [0, 1]$$

- Para cada elemento de I, la imagen es el grado de pertenencia de dicho ítem
- Para un subconjunto  $I_0 \subseteq I$ ,

$$\tau(I_0) = \min\{\tau(i) \text{ tal que } i \in I_0\}$$

- Un (fuzzy transactional) FT-set es un conjunto (crisp) de transacciones difusas

# Reglas difusas

$I = \{(\text{atributo}, \text{etiqueta difusa})\}$

Tupla  $t_i \rightarrow$  Transacción difusa  $\tilde{\tau}_i$   
 $\tilde{\tau}_i((Attr, Label)) = Label(t_i[Attr])$

	Peso	Altura
$t_1$	70	170
$t_2$	68	180
$t_3$	60	175
$t_4$	90	175
$t_5$	50	195

$\Rightarrow$

$\text{Label(Altura)} = \{\text{bajo, medio, alto, muy alto}\}$   
 $\text{Label(Peso)} = \{\text{ligero, medio, pesado, muy pesado}\}$

	$\tilde{\tau}_1$	$\tilde{\tau}_2$	$\tilde{\tau}_3$	$\tilde{\tau}_4$	$\tilde{\tau}_5$
<b>(Altura,bajo)</b>	0	1	0	0	0
<b>(Altura,medio)</b>	0	0.8	1	0	1
<b>(Altura,alto)</b>	1	0	1	0.5	0.5
<b>(Altura,m.alto)</b>	0.5	0	0	1	0
<b>(Peso,ligero)</b>	0	0	0	0	0
<b>(Peso,medio)</b>	1	1	0.5	0.5	0.25
<b>(Peso,pesado)</b>	0.5	1	1	1	1
<b>(Peso,m.pesado)</b>	0	0	0	0	0.25

# Reglas difusas

- Una regla de asociación difusa es una regla de asociación en un FT-set
- Una regla de asociación difusa  $A \rightarrow B$  tiene máximo cumplimiento en T si, y sólo si,

$$\tau(A) \leq \tau(B) \text{ para todo } \tau \in T$$

- Una regla de asociación es un caso particular de regla de asociación difusa

# Reglas difusas

- $(\text{Altura,bajo}) \Rightarrow (\text{Peso,medio})$  presenta total cumplimiento en  $T$
- $(\text{Altura,m.alto}) \Rightarrow (\text{Peso,m.pesado})$  no.

	$\tilde{\tau}_1$	$\tilde{\tau}_2$	$\tilde{\tau}_3$	$\tilde{\tau}_4$	$\tilde{\tau}_5$
<b>(Altura,bajo)</b>	0	1	0	0	0
<b>(Altura,medio)</b>	0	0.8	1	0	1
<b>(Altura,alto)</b>	1	0	1	0.5	0.5
<b>(Altura,m.alto)</b>	0.5	0	0	1	0
<b>(Peso,ligero)</b>	0	0	0	0	0
<b>(Peso,medio)</b>	1	1	0.5	0.5	0.25
<b>(Peso,pesado)</b>	0.5	1	1	1	1
<b>(Peso,m.pesado)</b>	0	0	0	0	0.25

# Reglas difusas

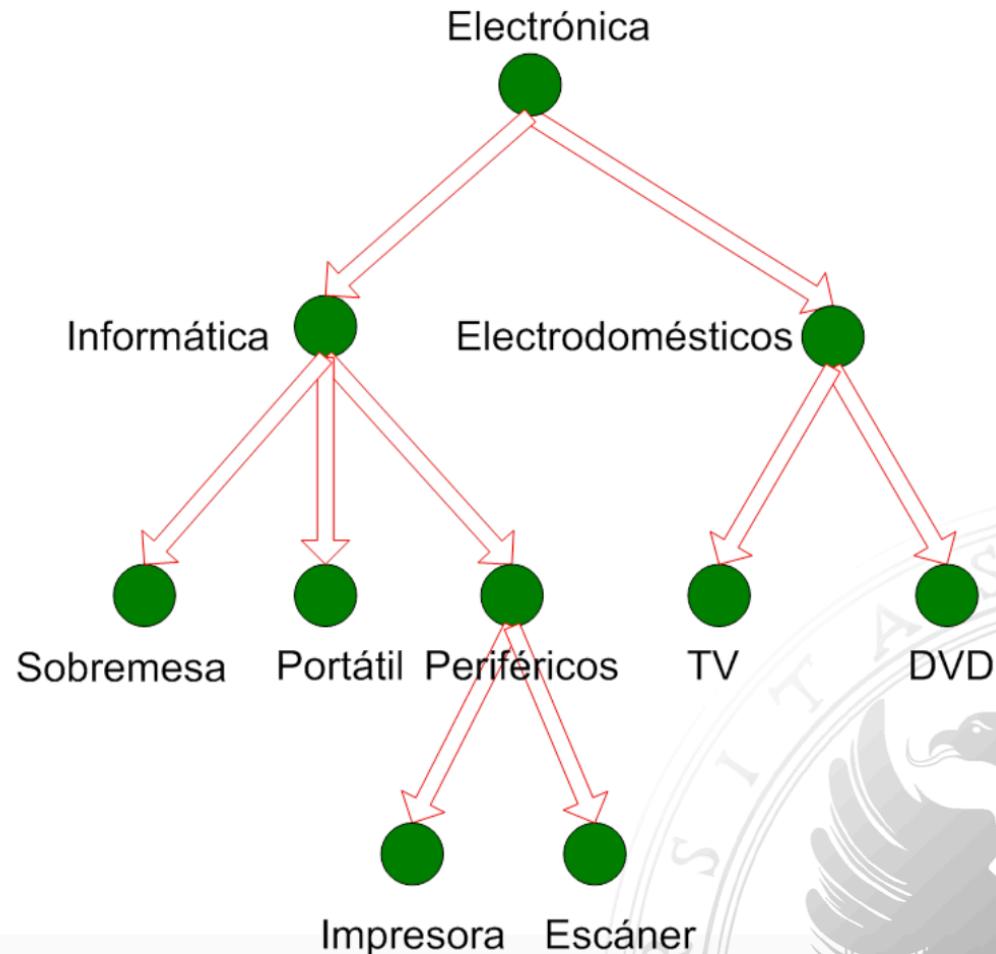
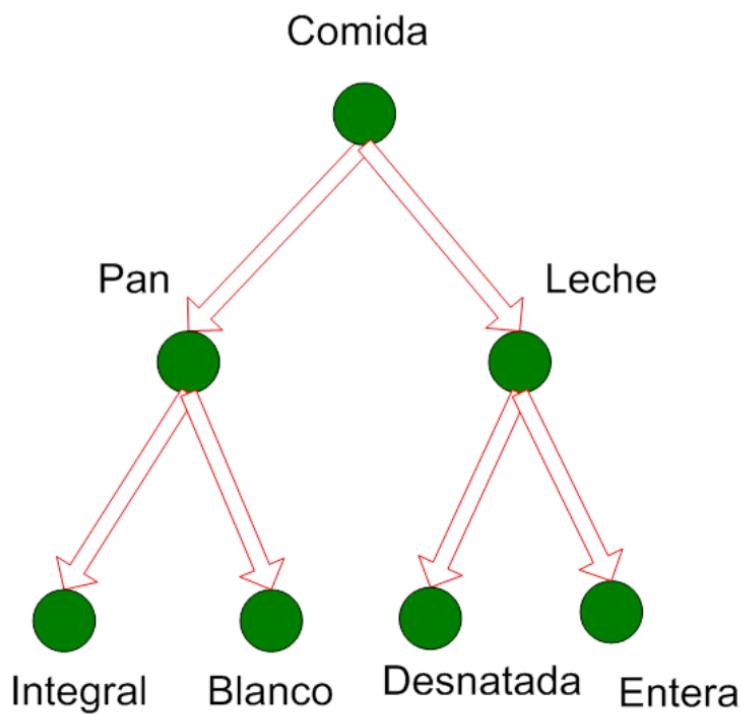
Algunos problemas:

- ❑ Diferentes enfoques para medir soporte y confianza
- ❑ Diferentes enfoques para medir factor de certeza (Problemas de Cardinalidad)
- ❑ Otros esquemas de valoración de Reglas
- ❑ Obtención de etiquetas
- ❑ Taxonomía de etiquetas (jerarquización)
- ❑ Desarrollo de Algoritmos
  
- ❑ Trabajo evaluable (3 personas, 30 minutos). **Reglas de asociación difusas.** Bibliografía:
  - Miguel Delgado, Nicolás Marín, Daniel Sánchez, and María-Amparo Vila, Fuzzy Association Rules: General Model and Applications, IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 11, NO. 2, APRIL 2003 , 214-225.

# Índice

- Concepto de regla de asociación
- Algoritmo Apriori
- Variaciones del algoritmo A priori
- Algoritmo FP-Growth
- Generación de reglas
- Evaluación de reglas
- Reglas difusas
- Otros aspecto

# Reglas multinivel



# Reglas multinivel

Por qué utilizar jerarquías de conceptos?

- Porque las reglas que involucran artículos en los niveles más bajos puede que no tengan soporte suficiente como para aparecer en algún patrón frecuente
- Porque las reglas a niveles bajos de la jerarquía son demasiado específicas

leche desnatada → pan blanco

leche entera → pan integral

leche desnatada → pan integral

indican una asociación entre pan y leche.

# Reglas negativas

En ocasiones, interesa obtener información del tipo

leche entera → **No** pan integral

Coca-Cola → **No** Pepsi

**No** Café → Té

- Estas reglas reciben el nombre de negativas
- Están formadas por ítem negativos o itemset negativos

# Reglas negativas

Una solución:

- ❑ Añadir los items negativos y aplicar los algoritmos conocidos

ID	Cesta
001	Pan, leche, huevos, <b>No pañales, No cerveza</b>
010	Pan, pañales, cerveza, <b>No leche, No huevos</b>
011	Leche, pañales, cerveza, <b>No huevos, No pan</b>
100	Pan, leche, pañales, cerveza, <b>No huevos</b>
101	Pan, leche, huevos, cerveza, <b>No pañales</b>

No es una buena idea,

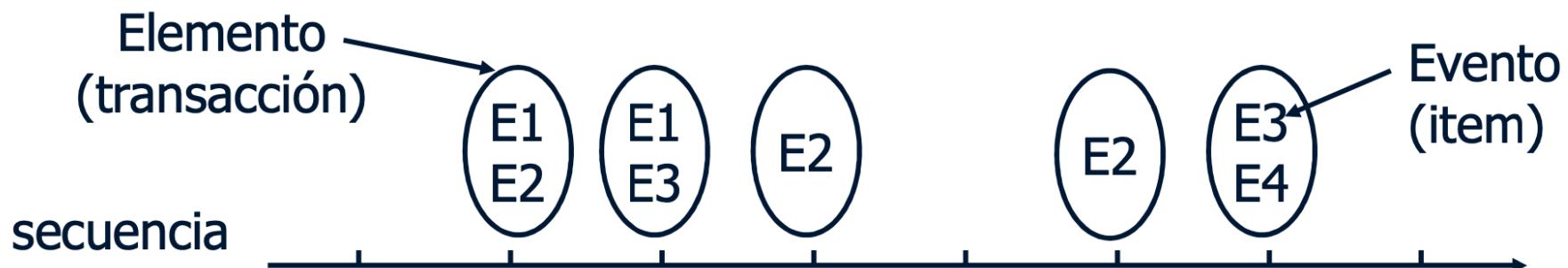
- el soporte de los items negativos es normalmente enorme (lo que no compramos) y van a salir reglas poco útiles
- Además, las cestas de la compra se vuelven enormes y la complejidad computacional no es asumible

# Reglas negativas

Otras soluciones:

- Lift.** Para lift negativos, los ítems están negativamente correlacionados
- Test estadísticos** para ver la correlación entre ítems. Por ejemplo,  $\chi^2$
- Medidas de interés** de reglas, en conjunción con una búsqueda guiada por cierto conocimiento.
- Adaptación de algoritmos existentes** para reglas negativas

# Análisis de secuencias



Base de datos	Secuencia	Elemento (Transacción)	Evento (Item)
Clients	Historial de compras de un cliente determinado	Conjunto de artículos comprados por un cliente en un instante concreto	Libros, productos...
Web	Navegación de un visitante del sitio web	Colección de ficheros vistos por el visitante tras un único click de ratón	Página inicial, información de contacto, fotografía...
Eventos	Eventos generados por un sensor	Eventos generados por un sensor en un instante t	Tipos de alarmas generadas
Genoma	Secuencia de ADN	Elemento de la secuencia de ADN	Bases A,T,G,C

# Análisis de secuencias

## Definición formal

Dado un conjunto de ítems I, una secuencia  $S = \langle A_1, A_2, \dots, A_n \rangle$  es una sucesión de itemsets de I

Dadas dos secuencias  $S = \langle A_1, A_2, \dots, A_n \rangle$  y  $T = \langle B_1, B_2, \dots, B_m \rangle$ , diremos que S está contenida en T si existen una sucesión de enteros  $i_1 < i_2 < \dots < i_n$  tales que  $A_j \subseteq B_{i_j}$  para todo j

Secuencia	Subsecuencia	¿Incluida?
$\langle \{2,4\} \{3,5,6\} \{8\} \rangle$	$\langle \{2\} \{3,5\} \rangle$	Sí
$\langle \{1,2\} \{3,4\} \rangle$	$\langle \{1\} \{2\} \rangle$	No
$\langle \{2,4\} \{2,4\} \{2,5\} \rangle$	$\langle \{2\} \{4\} \rangle$	Sí

# Análisis de secuencias

El soporte de una subsecuencia S se define como la fracción de secuencias de la base de datos que incluyen la subsecuencia S

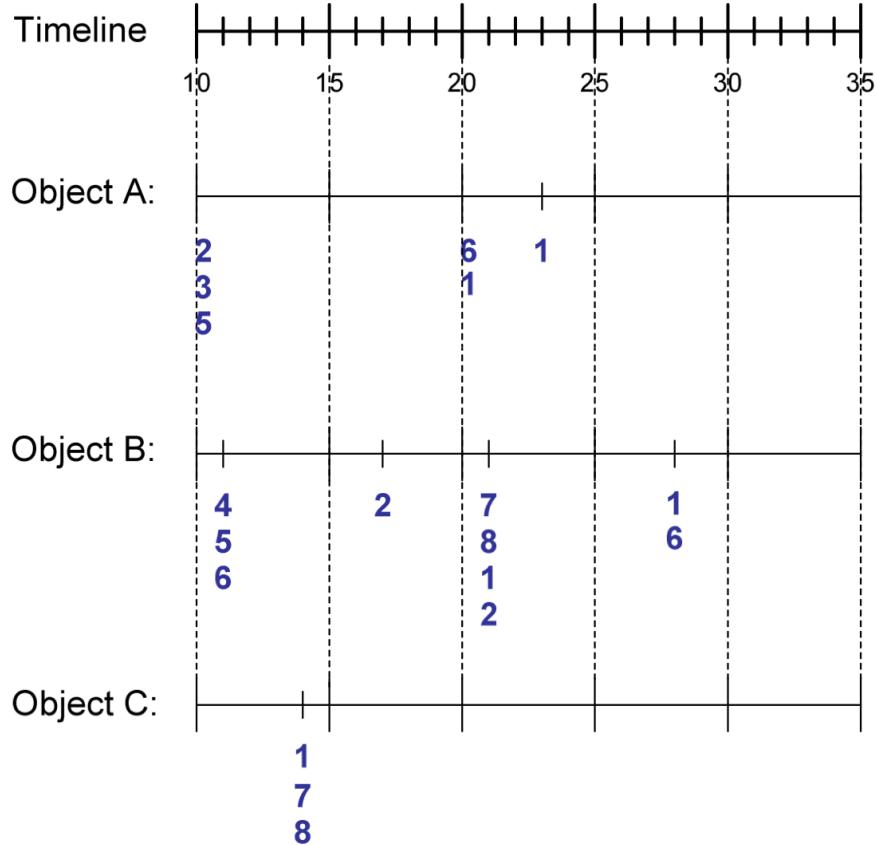
Un patrón secuencial es una secuencia con un soporte mayor que cierto soporte mínimo

Normalmente, se adaptan los algoritmos de cálculo de itemsets frecuentes para cálculo de patrones secuenciales (por ejemplo, AprioriAll)

# Análisis de secuencias

Object	Timestamp	Events
A	10	2, 3, 5
A	20	6, 1
A	23	1
B	11	4, 5, 6
B	17	2
B	21	7, 8, 1, 2
B	28	1, 6
C	14	1, 7, 8

Base de datos  
de secuencias



# Análisis de secuencias

Object	Timestamp	Events
A	1	1,2,4
A	2	2,3
A	3	5
B	1	1,2
B	2	2,3,4
C	1	1, 2
C	2	2,3,4
C	3	2,4,5
D	1	2
D	2	3, 4
D	3	4, 5
E	1	1, 3
E	2	2, 4, 5

*MinSupp = 50%*

**Ejemplos de subsecuencias frecuentes:**

- |                 |       |
|-----------------|-------|
| < {1,2} >       | s=60% |
| < {2,3} >       | s=60% |
| < {2,4}>        | s=80% |
| < {3} {5}>      | s=80% |
| < {1} {2} >     | s=80% |
| < {2} {2} >     | s=60% |
| < {1} {2,3} >   | s=60% |
| < {2} {2,3} >   | s=60% |
| < {1,2} {2,3} > | s=60% |

# Trabajos evaluables

- ❑ (3 personas, 30 minutos). Análisis de secuencias. Algoritmo AprioriAll.  
Bibliografía:
  - R. Agrawal and R. Srikant, Mining sequential patterns, Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, 1995, pp. 3-14.
  - Introducción a la Minería de Datos. José Hernández Orallo, M.José Ramírez Quintana, Cèsar Ferri Ramírez. Pearson, 2004. Capítulo 9.
- ❑ (3 personas, 30 minutos) Algoritmos ECLAT, DECLAT, FP-Growth.  
Ejemplos. Bibliografía: J. Han, M. Kamber and J. Pei. *Data Mining. Concepts and Techniques*. Morgan Kaufmann. 2012. Chapter 6
- ❑ (3 personas, 30 minutos) Estrategias para encontrar reglas negativas.  
Ejemplos. Bibliografía:
  - C. C. Aggarwal and J. Han, *Frequent Patter Mining*, Springer, 2014. Chapter 6 (y referencias en ese capítulo)
  - C. Zhang and S. Zhang, *Association Rules Mining. Models and Algorithms*, Lecture Notes in Artificial Intelligence 2307, Springer, 2002

# Bibliografía

- ❑ Introducción a la Minería de Datos. José Hernández Orallo, M.José Ramírez Quintana, Cèsar Ferri Ramírez. Pearson, 2004. Capítulo 9.
- ❑ J. Han, M. Kamber and J. Pei. Data Mining, Second Edition: Concepts and Techniques. Morgan Kaufmann, 2006. Capítulos 6,7.
- ❑ Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, May 2014. Capítulos 8,9,10.
- ❑ P-N Tan, M. Steinbach & V. Kumar: Introduction to Data Mining Addison-Wesley, 2006. capítulos 6, 7
- ❑ C. Aggarwal, Data Mining: The textbook, Springer, 2015.
- ❑ C. Zhang, S. Zhang. Association rule mining : models and algorithms. Springer, 2002

Algunas transparencias y gráficos tomados de:

- <http://elvex.ugr.es/idbis/dm/>

# Bibliografía

- ❑ Agrawal & Skirant, Fast Algorithms for Mining Association Rules, VLDB'94.
- ❑ Park, Chen & Yu, An Effective Hash-Based Algorithm for Mining Association Rules, SIGMOD'95 (DHP)
- ❑ Toivonen, Sampling Large Databases for Association Rules, VLDB'96.
- ❑ Park, Yu & Chen, Mining Association Rules with Adjustable Accuracy, CIKM'97
- ❑ Brin, Motwani, Ullman & Tsur, Dynamic itemset counting and implications rules for market basket data, SIGMOD'97 (DIC)
- ❑ Savasere, Omiecinski & Navathe: An Efficient Algorithm for Mining Association Rules in Large Databases, VLDB'95
- ❑ Berzal, Cubero, Sánchez & Serrano: TBAR: An efficient method for association rule mining in relational databases, Data & Knowledge Engineering, 2001
- ❑ Han, Pei & Yin, Mining Frequent Patterns without Candidate Generation, SIGMOD'2000 (FP-Growth)
- ❑ Berzal, Blanco, Sánchez & Vila, Measuring the accuracy and interest of association rules: A new framework, Intelligent Data Analysis, 2002
- ❑ Hilderman & Hamilton: Evaluation of interestingness measures for ranking discovered knowledge, PAKDD'2001
- ❑ Tan, Kumar & Srivastava, Selecting the right objective measure for association analysis. Information Systems, vol. 29, pp. 293-313, 2004