



# Universidad Autónoma de Nuevo León

## Facultad de Ciencias Físico Matemáticas



**Semestre: 7**

**Materia:**

Minería de datos

**Maestra:**

Mayra Cristina Berrones Reyes

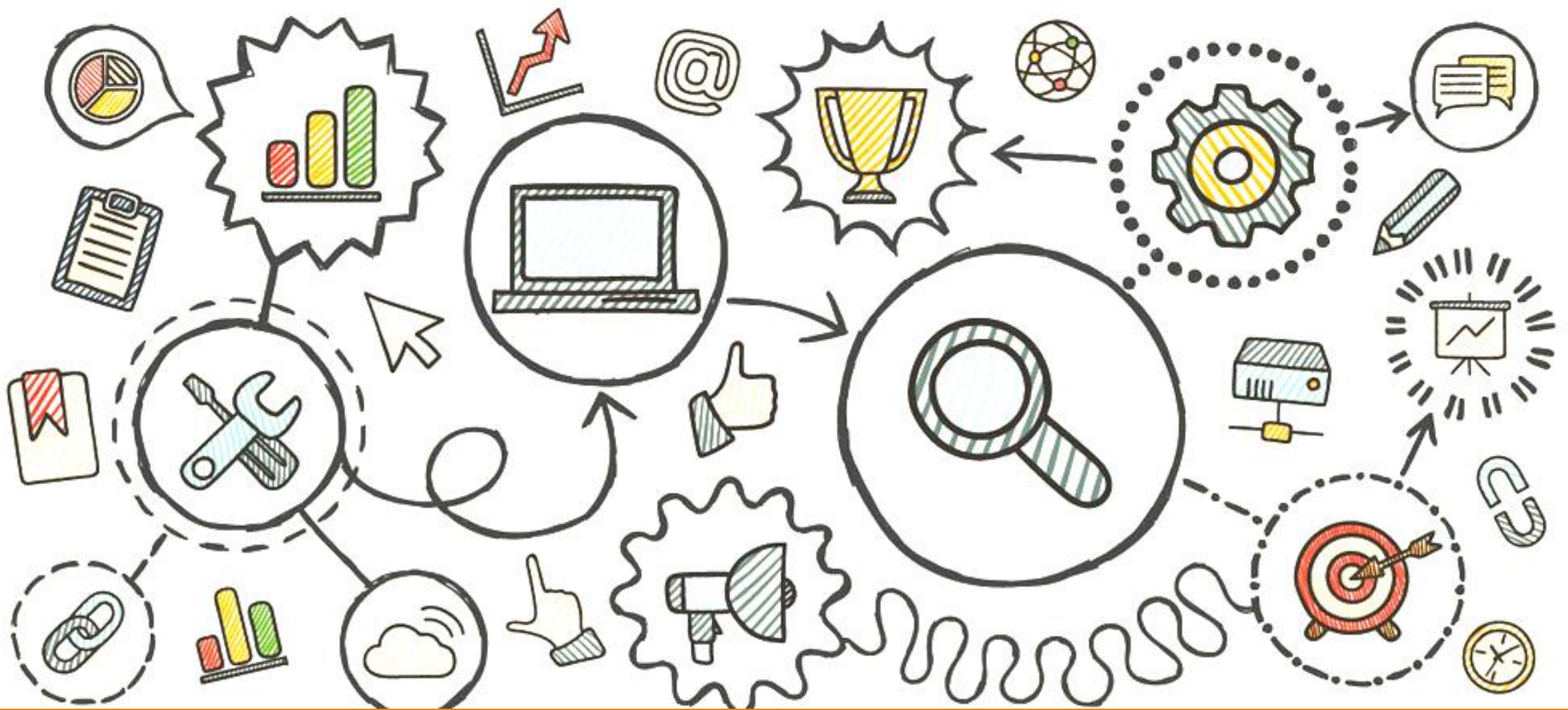
**Alumnos :**

Andrea López Solís	#1822031
Daniela Govea Serna	#1722714
Francisco García Sánchez Armáss	#1816358
Jesús Eduardo Valencia González	#1630606
Karyme Mayela Gauna Rodríguez	#1819032

Grupo: 003

Aula: AVI2

**San Nicolas de los Garza Nuevo León a 18 de septiembre del 2020**



# Reglas de Asociación

(Association Rules)



# Reglas de asociación

Búsqueda de patrones frecuentes, asociaciones, correlaciones o estructuras causales entre conjuntos de elementos u objetos en bases de datos de transacciones, bases de datos relacionales y otros repositorios de información disponibles.

## Aplicaciones

- Análisis de datos de la banca.
- Cross-marketing (poner la crema batida junto a las fresas).
- Diseño de catálogos.



# Ejemplo

Dado un conjunto de transacciones, encontrar reglas que predigan la ocurrencia de un artículo según las ocurrencias de otros artículos en la transacción.

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

Ejemplos:

- La gente que comprara pan, comprara también leche.
- La gente que comprara soda, comprara también papas de funda.
- La gente que comprara pan, comprara también mermelada.

**Soporte:**

Fracción de transacciones que contiene un itemset.

$$s(\{\text{Leche, Pan}\}) = 3$$

$$s(\{\text{Soda, Chips}\}) = 4$$

**Conjunto de elementos frecuente:**

Un conjunto de elementos cuyo soporte es mayor o igual que un umbral de mínimo.

### Conjunto de elementos:

Una colección de uno o más artículos, por ejemplo, {leche, pan, mermelada}. k-itemset, un conjunto de elementos que contiene k elementos.

### Recuento de soporte:

Frecuencia de ocurrencia de un itemset.

$$S(\{\text{Leche, Pan}\}) = 3$$

$$S(\{\text{Soda, Chips}\}) = 4$$

### Confianza (c):

Mide que tan frecuente items en Y aparecen en transacciones que contienen X.

$$s = \frac{\sigma(\{\text{Leche, Pan}\})}{\text{\#Núm. transiciones}} = \frac{3}{8} = 0.375 \quad c = \frac{\sigma(\{\text{Leche, Pan}\})}{\sigma(\{\text{Pan}\})} = \frac{3}{4} = 0.75$$

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

# El Objetivo

---

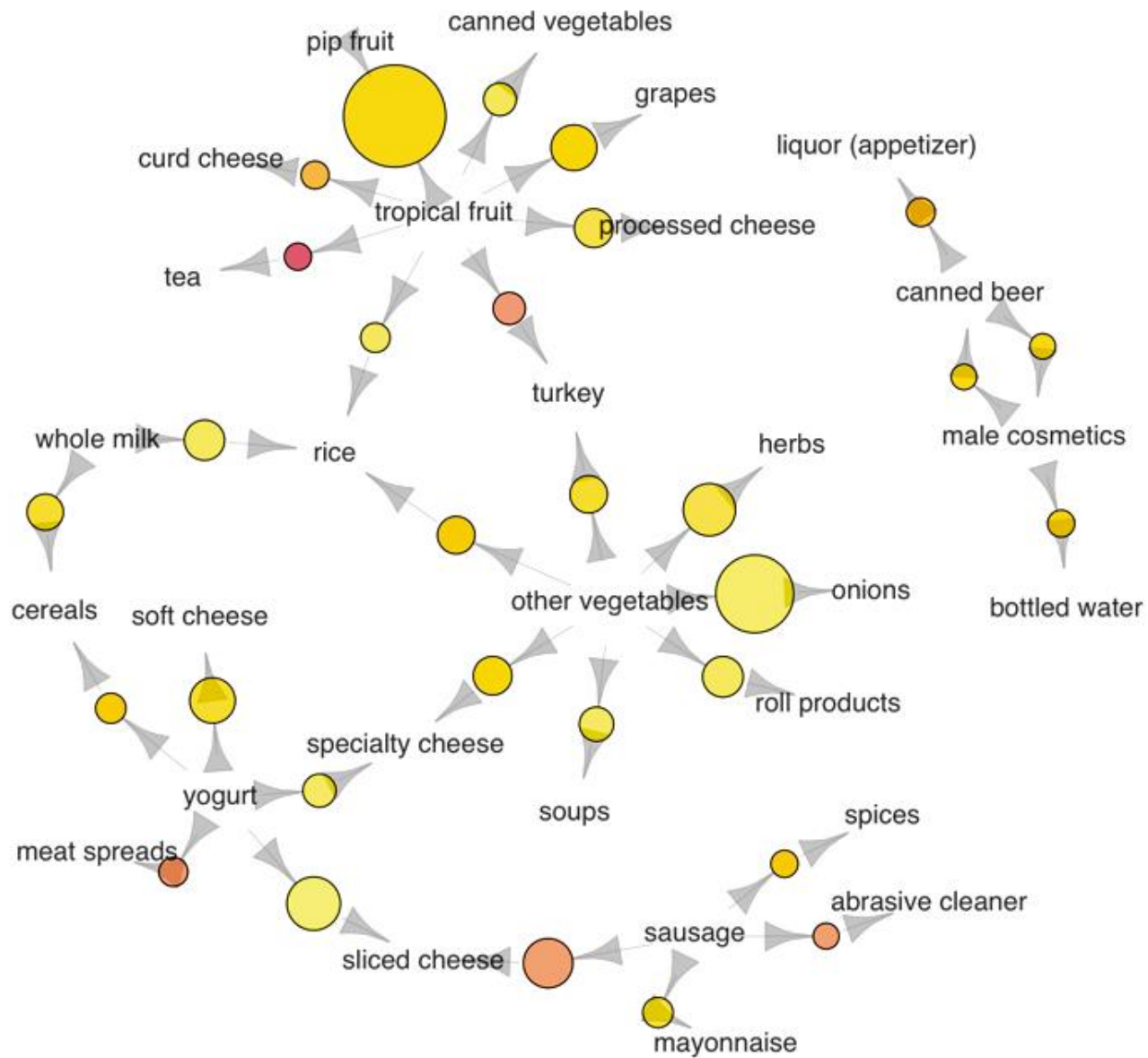
Dado un conjunto de transacciones  $T$ , el objetivo de la minería de reglas de asociación es encontrar todas las reglas teniendo:

- Umbral mínimo de soporte
- Umbral mínimo de confianza

## **Enfoque de fuerza bruta:**

- Lista todas las reglas de asociación posibles.
- Compruebe el soporte y la confianza para cada regla .
- Elimine las reglas que fallan en los umbrales mínimos.





# Reglas de la Asociación Minera: Enfoque de dos pasos

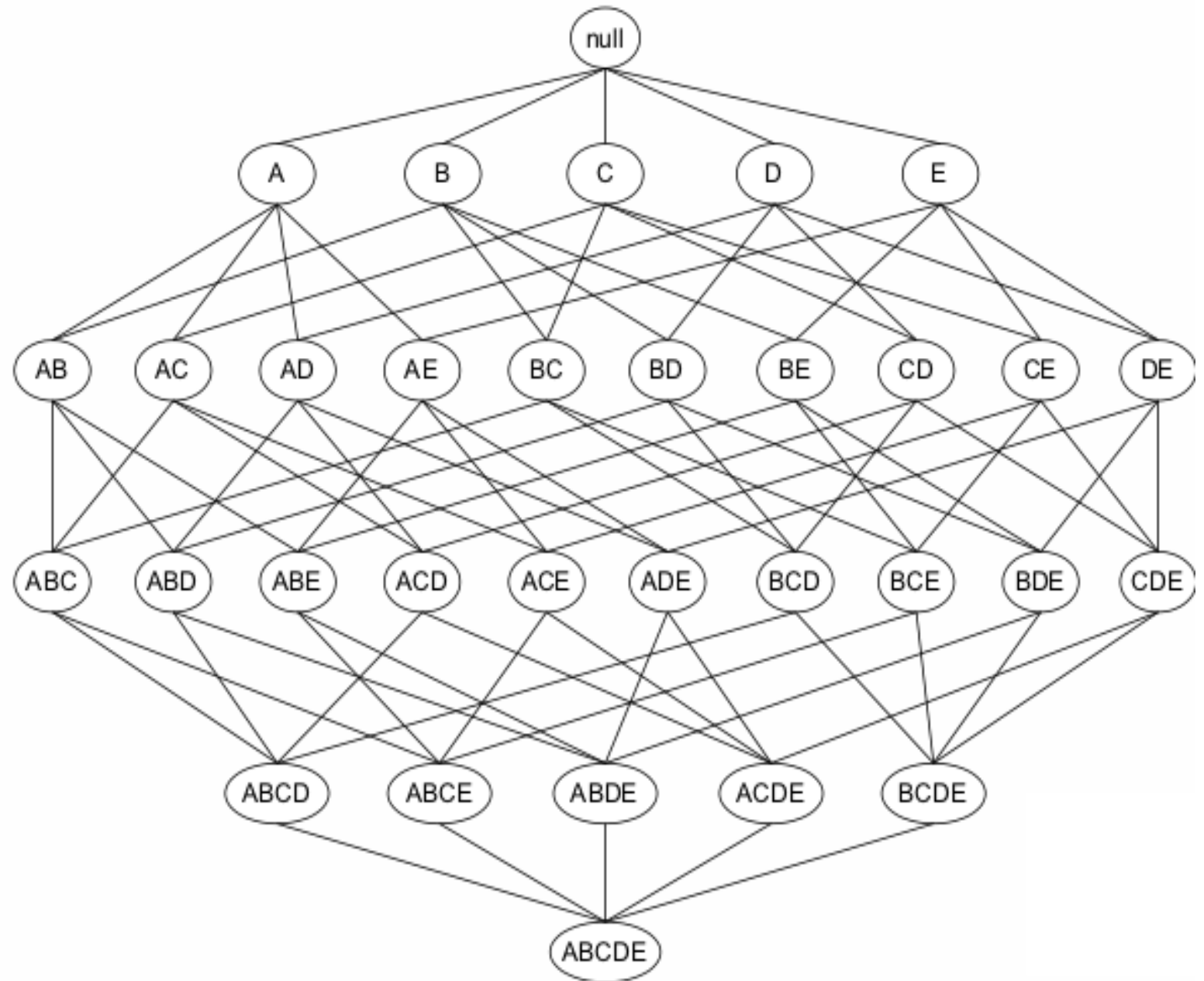
# RAM: Enfoque 2 pasos

## Generación de elementos frecuentes:

Generar todos los conjuntos de elementos cuyo soporte  $\geq$  minsup.

## Generación de reglas:

Generar reglas de alta confianza a partir de un conjunto de elementos frecuentes. Cada regla es una partición binaria de un conjunto de elementos frecuente.





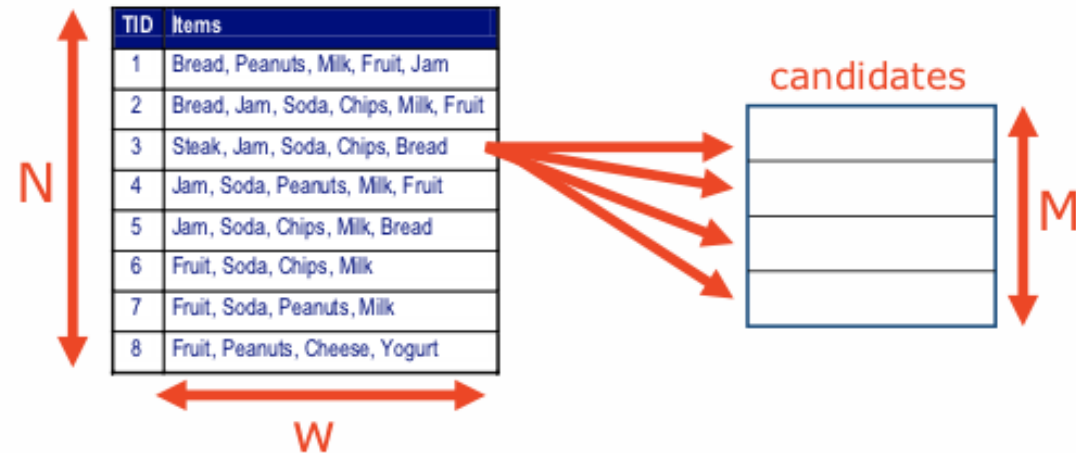
Cada conjunto de elementos en la red es un conjunto de elementos frecuente candidato.

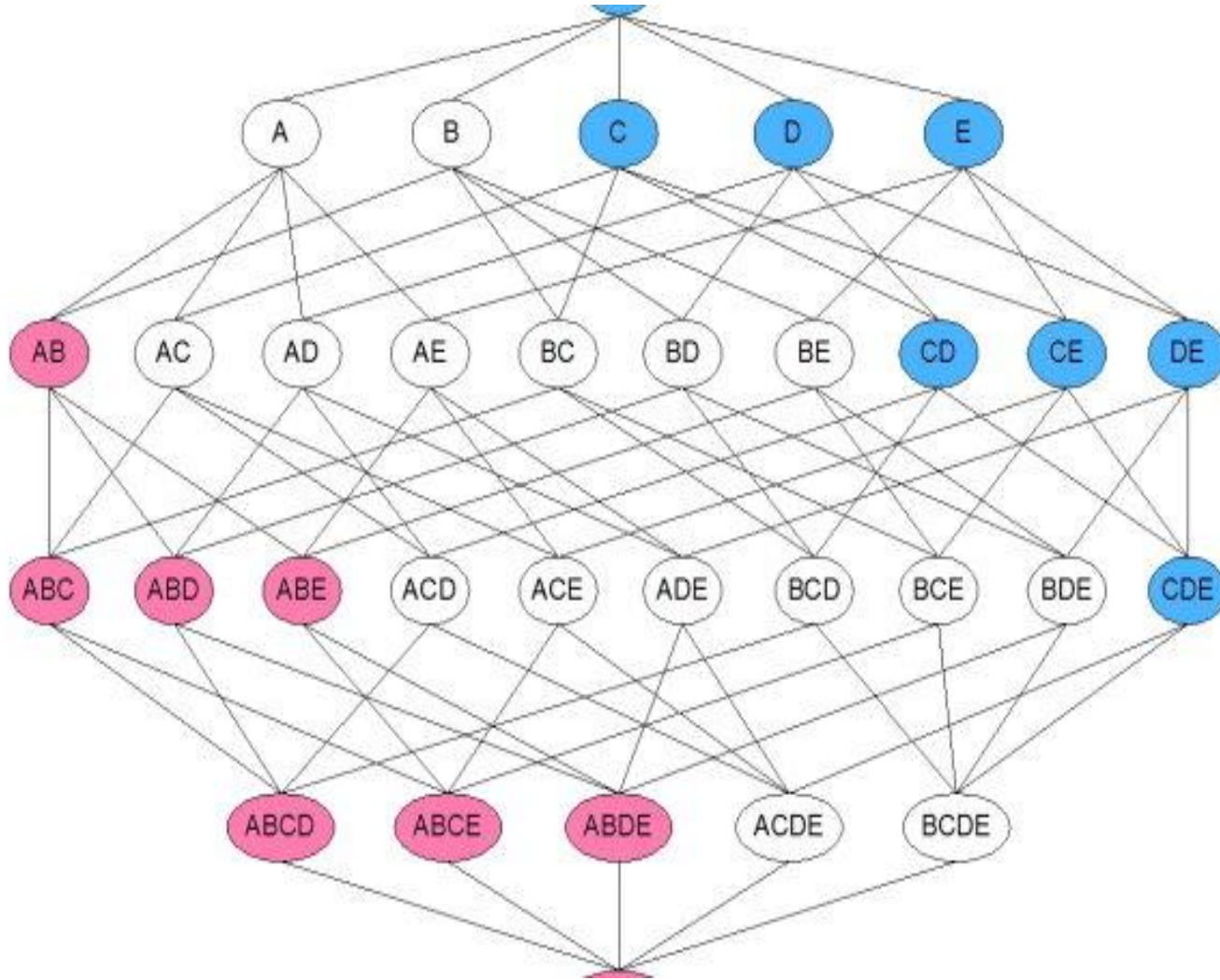
- Calcular el soporte de cada candidato escaneando la base de dato
  - Empareja cada transacción con cada candidato

---

### Estrategias de generación de elementos frecuentes

- Reducir el número de candidatos (M)
  - Reducir el número de transacciones (N)
  - Reducir el número de comparaciones (NM)
- 





Reglas de la  
Asociación:  
Principio “apriori”

# Reduciendo el número de candidatos

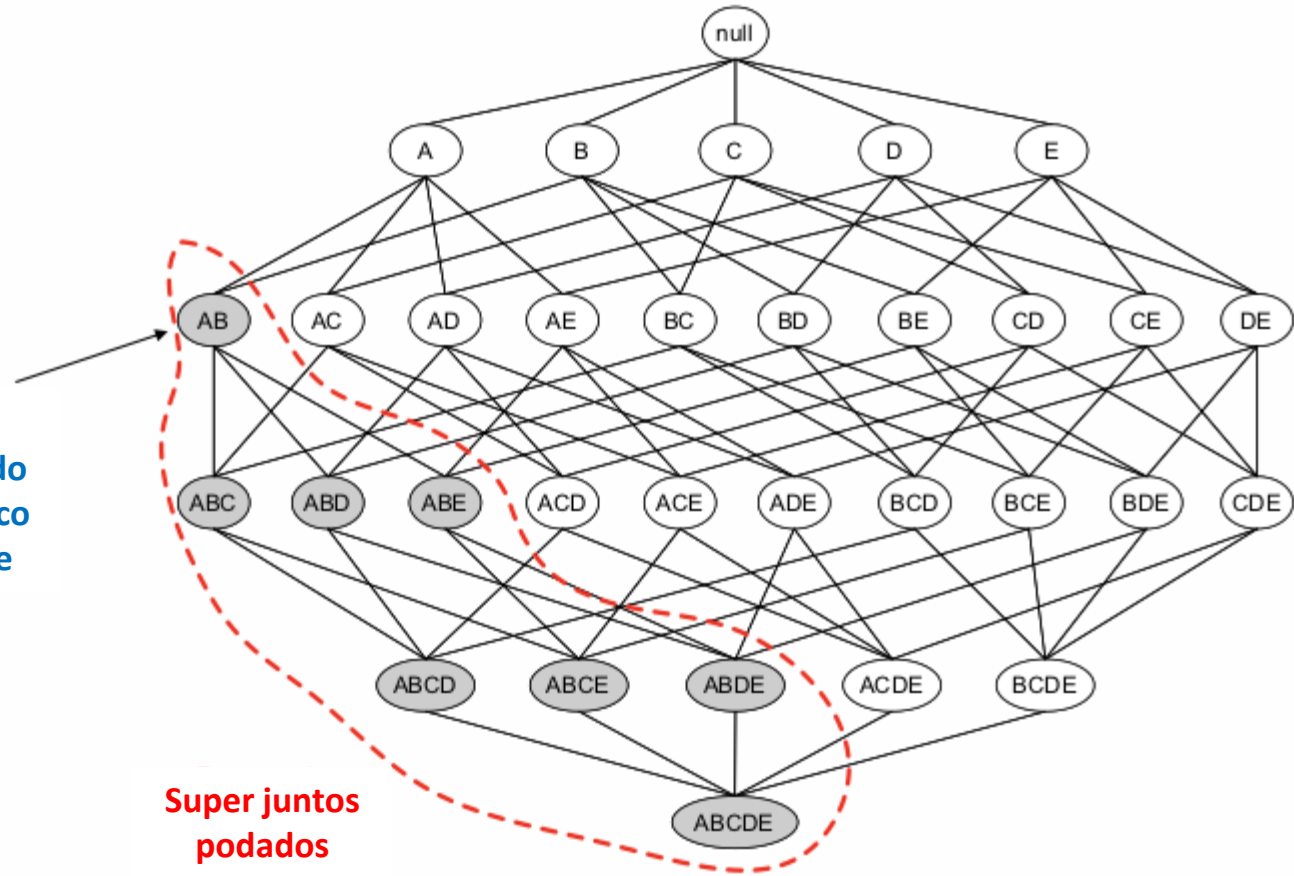
**Principio de Apriori:** si un conjunto de elementos es frecuente, entonces todos sus subconjuntos también deben ser frecuentes. El principio de Apriori se mantiene debido a la siguiente propiedad de la medida de soporte:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

El soporte de un conjunto de elementos nunca excede el soporte de sus subconjuntos. Esto se conoce como la propiedad anti-monótona de soporte

Se ha  
encontrado  
que es poco  
frecuente

Super juntos  
podados



# Algoritmo Apriori

---

El núcleo del algoritmo de Apriori.

- Utilizar los conjuntos frecuentes (k-1) para generar candidatos a k-items frecuentes.
- Utilizar el escaneo de la base de datos y la coincidencia de patrones para recoger los recuentos de los conjuntos de elementos candidatos.

Comprimir una gran base de datos en una estructura compacta de árbol de patrones frecuentes (FP-tree). Muy condensado, pero completo para la minería de patrones frecuentes. Evita costosos análisis de bases de datos.

Utilice un método de minería de patrones frecuentes, basado en el árbol de FP. Una metodología de dividir y conquistar: descomponer las tareas de minería en los más pequeños.

# Algoritmo Apriori

**APRIORI** ( $\mathbf{D}, \mathcal{I}, \text{minsup}$ ):

```
1  $\mathcal{F} \leftarrow \emptyset$ 
2  $\mathcal{C}^{(1)} \leftarrow \{\emptyset\}$  // Initial prefix tree with single items
3 foreach  $i \in \mathcal{I}$  do Add  $i$  as child of  $\emptyset$  in  $\mathcal{C}^{(1)}$  with  $\text{sup}(i) \leftarrow 0$ 
4  $k \leftarrow 1$  //  $k$  denotes the level
5 while  $\mathcal{C}^{(k)} \neq \emptyset$  do
6   COMPUTESUPPORT ( $\mathcal{C}^{(k)}, \mathbf{D}$ )
7   foreach leaf  $X \in \mathcal{C}^{(k)}$  do
8     if  $\text{sup}(X) \geq \text{minsup}$  then  $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, \text{sup}(X))\}$ 
9     else remove  $X$  from  $\mathcal{C}^{(k)}$ 
10   $\mathcal{C}^{(k+1)} \leftarrow \text{EXTENDPREFIXTREE}(\mathcal{C}^{(k)})$ 
11   $k \leftarrow k + 1$ 
12 return  $\mathcal{F}^{(k)}$ 
```

**COMPUTESUPPORT** ( $\mathcal{C}^{(k)}, \mathbf{D}$ ):

```
1 foreach  $\langle t, \mathbf{i}(t) \rangle \in \mathbf{D}$  do
2   foreach  $k$ -subset  $X \subseteq \mathbf{i}(t)$  do
3     if  $X \in \mathcal{C}^{(k)}$  then  $\text{sup}(X) \leftarrow \text{sup}(X) + 1$ 
```

**EXTENDPREFIXTREE** ( $\mathcal{C}^{(k)}$ ):

```
1 foreach leaf  $X_a \in \mathcal{C}^{(k)}$  do
2   foreach leaf  $X_b \in \text{SIBLING}(X_a)$ , such that  $b > a$  do
3      $X_{ab} \leftarrow X_a \cup X_b$ 
4     // prune candidate if there are any infrequent subsets
5     if  $X_j \in \mathcal{C}^{(k)}$ , for all  $X_j \subset X_{ab}$ , such that  $|X_j| = |X_{ab}| - 1$  then
6       if no extensions from  $X_a$  then
7         remove  $X_a$ , and all ancestors of  $X_a$  with no extensions, from  $\mathcal{C}^{(k)}$ 
8 return  $\mathcal{C}^{(k)}$ 
```

## Ejemplo

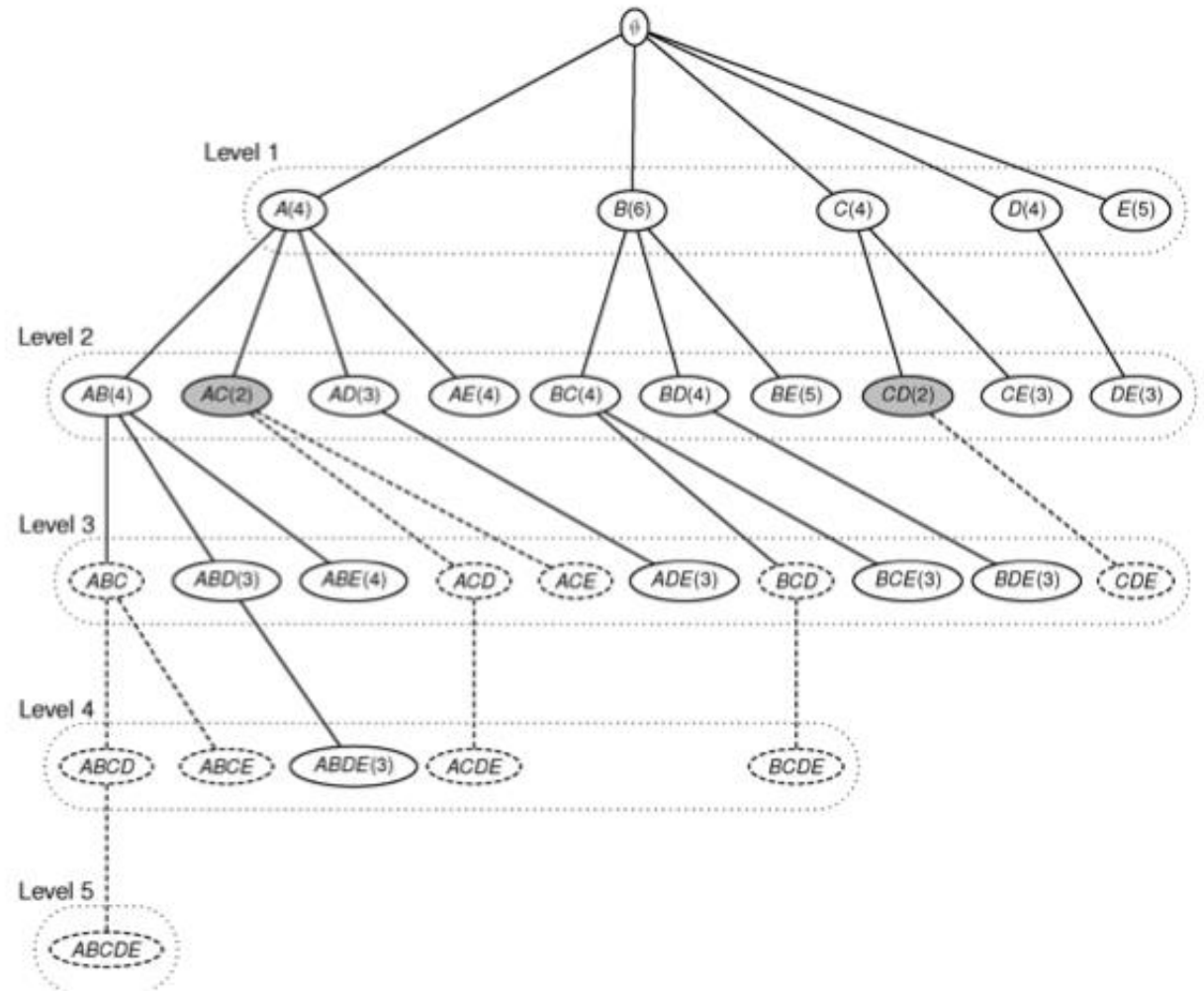
Dada la siguiente base de datos y un soporte mínimo de 3, genere todos los conjuntos de elementos frecuentes.

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

Base de Datos Binaria

t	i(t)
1	ABDE
2	BCE
3	ABDE
4	ABCE
5	ABCDE
6	BCD

Base de Datos Transaccional





## Ejercicio de practica

Se procede a identificar los *itemsets* frecuentes y, a partir de ellos, crear reglas de asociación.

---

### Transacción

{A, B, C, D}

{A, B, D}

{A, B}

{B, C, D}

{B, C}

{C, D}

{B, D}

---

Para este problema se considera que un *item* o *itemset* es frecuente si aparece en un mínimo de 3 transacciones, es decir, su soporte debe de ser igual o superior a  $3/7 = 0.43$ . Se inicia el algoritmo identificando todos los *items* individuales (*itemsets* de un único *item*) y calculando su soporte.

---

### Itemset (k=1)

### Ocurrencias

### Soporte

{A}

{B}

{C}

{D}

---

A continuación, se generan todos los posibles *itemsets* de tamaño  $k = 2$  que se pueden crear con los *itemsets* que han superado el paso anterior y se calcula su soporte.

Item k=2

Los *itemsets* \_\_\_\_\_superan el límite ( $\geq 0.43$ ) de soporte, por lo que son frecuentes. Los *itemsets* \_\_\_\_\_no superan el soporte mínimo ( $\leq 0.43$ ) por lo que se descartan.

Itemset (k=2)	Ocurrencias	Soporte
---------------	-------------	---------

Item k=2

Se repite el proceso, esta vez creando *itemsets* de tamaño  $k = 3$ .

Itemset (k=2)	Ocurrencias	Soporte
---------------	-------------	---------

Item k=3

Los *itemsets* \_\_\_\_\_contienen subconjuntos infrecuentes, por lo que son descartados. Para los restantes se calcula su soporte.

Itemset (k=3)	Ocurrencias	Soporte
---------------	-------------	---------

El *items* \_\_\_\_\_ no supera el soporte mínimo por lo que se considera infrecuente. Al no haber ningún nuevo *itemset* frecuente, se detiene el algoritmo.

Como resultado de la búsqueda se han identificado los siguientes *itemsets* frecuentes:

---

Itemset frecuentes

---

De todas las posibles reglas, únicamente:

- 
- 
- 

superan el límite de confianza.

---

Reglas

Confianza

Confianza

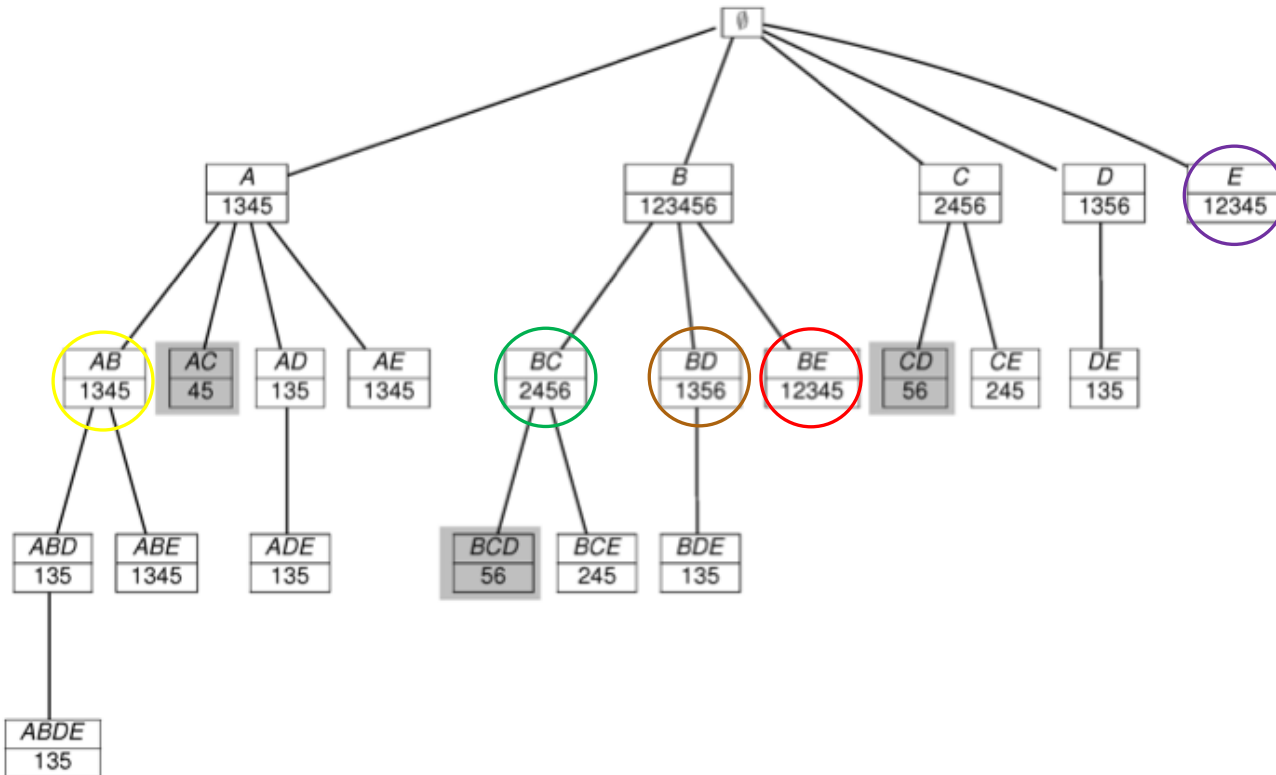
---

# Eclat (Ejemplo)

Dado  $t(X)$  y  $t(Y)$  para dos conjuntos de elementos frecuentes  $X$  e  $Y$ , entonces:

$$t(XY) = t(X) \wedge t(Y).$$

$$\text{sup}(XY) = |t(XY)|$$



D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

Binary Database

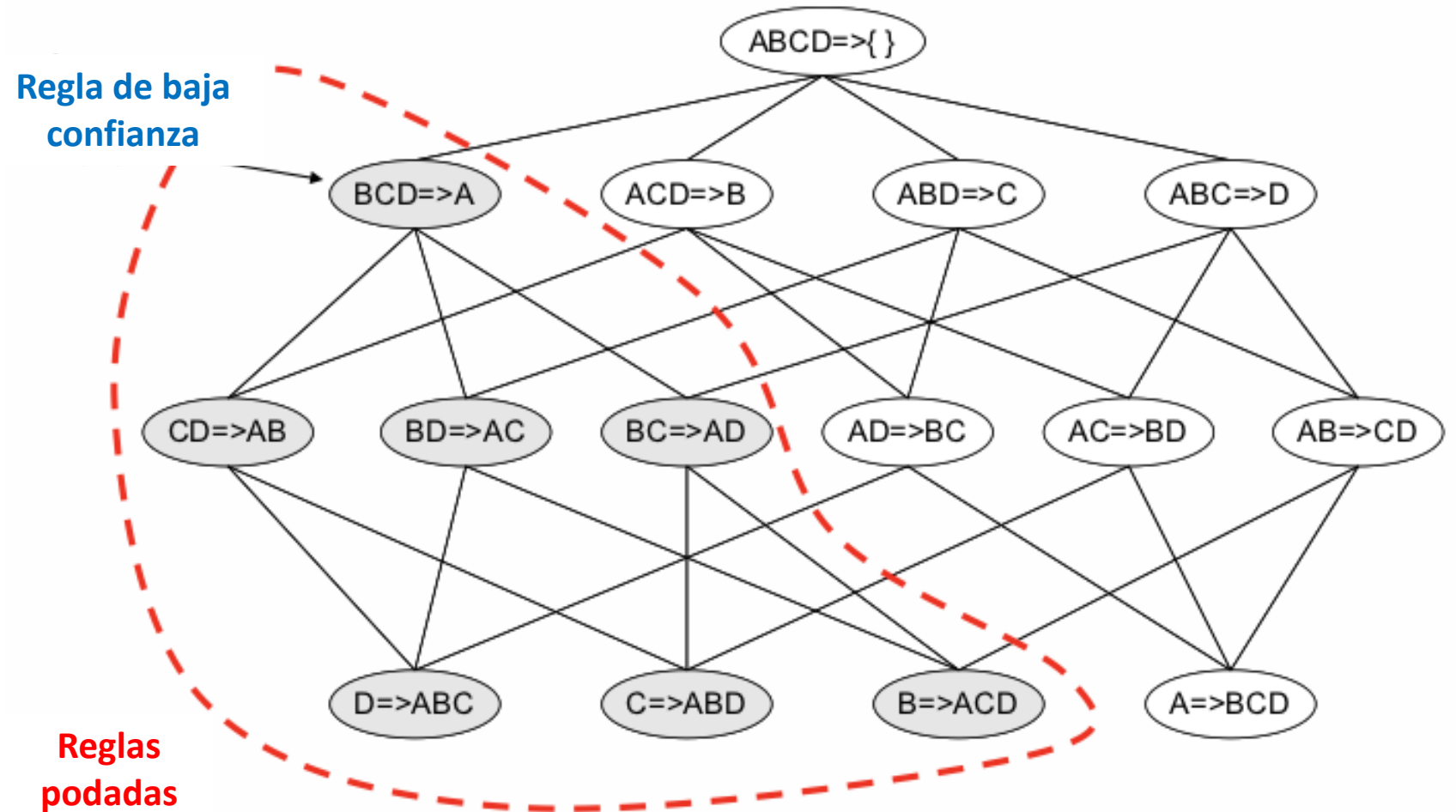
t	i(t)
1	ABDE
2	BCE
3	ABDE
4	ABCE
5	ABCDE
6	BCD

Transaction Database

t(x)				
A	B	C	D	E
1	1	2	1	1
3	2	4	3	2
4	3	5	5	3
5	4	6	6	4
5				5
6				

Vertical Database

¿Cómo generar reglas de manera eficiente a partir de elementos frecuentes?



# Generar Reglas

La confianza no tiene una propiedad anti-monótona.

$c(ABC \rightarrow D)$  puede ser mayor o menor que  $c(AB \rightarrow D)$ .

Pero la confianza en las reglas generadas desde el mismo conjunto de elementos tiene una propiedad antimonotónica.

$$\begin{aligned} L = \{A, B, C, D\}: c(BCD \rightarrow A) &= c(BC \rightarrow AD) = c(B \rightarrow ACD) \\ &= c(BD \rightarrow AC) = c(D \rightarrow ABC) \\ &= c(CD \rightarrow AB) = c(C \rightarrow ABD) \end{aligned}$$

$$\begin{aligned} L = \{A, B, C, D\}: c(ACD \rightarrow B) &= c(AC \rightarrow BD) = c(A \rightarrow BCD) \\ &= c(AD \rightarrow BC) = c(D \rightarrow ABC) \\ &= c(CD \rightarrow AB) = c(C \rightarrow ABD) \end{aligned}$$

$$\begin{aligned} L = \{A, B, C, D\}: c(ABD \rightarrow C) &= c(AD \rightarrow BC) = c(A \rightarrow BCD) \\ &= c(AB \rightarrow CD) = c(B \rightarrow ACD) \\ &= c(BD \rightarrow AC) = c(D \rightarrow ABC) \end{aligned}$$

$$\begin{aligned} L = \{A, B, C, D\}: c(ABC \rightarrow D) &= c(AB \rightarrow CD) = c(A \rightarrow BCD) \\ &= c(AC \rightarrow BD) = c(C \rightarrow ABD) \\ &= c(BC \rightarrow AD) = c(B \rightarrow ACD) \end{aligned}$$

La confianza es antimonotona con respecto al número de artículos en el lado derecho de la regla.



# Bibliografía

---

- JayWrkr (Mayo 5,2019). *Association Rules*. [en línea]. México. Disponible en: <https://medium.com/@jaywrkr/miner%C3%ADa-de-datos-3-f75d15f90c46> [2020, 12 septiembre]
- Ansel Yoan Rodríguez González, José Francisco Martínez Trinidad, Jesús Ariel Carrasco Ochoa, José Ruiz Shulcloper (Marzo 31,2009). *Minería de Reglas de Asociación sobre Datos Mezclados*. [en línea]. Puebla, México. Disponible en: <http://ccc.inaoep.mx/portalfiles/file/CCC-09-001.pdf> [2020, 12 septiembre]
- Amat Rodrigo Joaquín (Junio, 2018). *Reglas de asociación y algoritmo a priori con R*. [en línea]. México. Disponible en: [https://www.cienciadedatos.net/documentos/43\\_reglas\\_de\\_asociacion](https://www.cienciadedatos.net/documentos/43_reglas_de_asociacion) [2020, 14 septiembre]