

# Tema 5

## Aprendizaje inductivo. Adquisición de conceptos

Ciencias de la Computación e Inteligencia Artificial  
Universidad de Huelva

noviembre 2021

- 1 Introducción
- 2 Análisis de diferencias
  - Winston
    - Redes semánticas
    - Método de cotejamiento
    - Generalización
    - Especialización
- 3 Espacio de versiones
  - Find-S
  - List-Then-Eliminate
  - Eliminación de candidatos
- 4 Bibliografía

- 1 Introducción
- 2 Análisis de diferencias
  - Winston
    - Redes semánticas
    - Método de cotejamiento
    - Generalización
    - Especialización
- 3 Espacio de versiones
  - Find-S
  - List-Then-Eliminate
  - Eliminación de candidatos
- 4 Bibliografía

- El **aprendizaje inductivo** consiste en inducir información de una concepto a partir de un conjunto de cosas en concreto. No requiere información previa del dominio.
- Ejemplo: ese gato tiene 4 patas  $\rightarrow$  todos los gatos tienen 4 patas.
- Si veo 1 caso supongo que todos los casos son iguales hasta que encuentre una contradicción que me obligue a remodelar las informaciones.

## ■ Aprendizaje inductivo simbólico

- Utiliza una representación simbólica (redes semánticas, reglas, programación lógica, ...)

## ■ Aprendizaje inductivo subsimbólico

- Utiliza una representación subsimbólica (conjuntos difusos,...)
- Se desarrolla por medio de algoritmos de ajuste paramétrico

## Formas de aprendizaje inductivo simbólico

- Si los ejemplos reflejan situaciones con múltiples objetos y relaciones
  - **Adquisición de conceptos**
- Si los ejemplos se refieren a conjuntos atributo-valor
  - **Clasificación supervisada**
- Si se pretende adquirir un modelo lógico
  - **Programación Lógica Inductiva**

- 1 Introducción
- 2 **Análisis de diferencias**
  - **Winston**
    - Redes semánticas
    - Método de cotejamiento
    - Generalización
    - Especialización
- 3 Espacio de versiones
  - Find-S
  - List-Then-Eliminate
  - Eliminación de candidatos
- 4 Bibliografía

- Surge en la década de los 70
- Se produce un declive de la aproximación neuronal
- Aparecen los primeros lenguajes de manipulación simbólica
- La investigación se centra en problemas de juegos
- Se intentan obtener mecanismos de aprendizaje simbólico generales con muy poca información de partida
- **Algoritmos propuestos**
  - Winston (1970), Hayes-Roth (1977), Vere (1975), Michalski-Dietterich (1981)
  - Mitchell (1982) propone un marco unificado (espacio de versiones)



- Desarrollado a lo largo de la década de los 70
- Se considera el punto de partida del **Aprendizaje Basado en Similitudes** (SBL)
- El autor lo denomina **Learning by analyzing differences**
- Introduce el concepto de **quasi-ejemplo** (ejemplo negativo muy próximo a los ejemplos positivos)
- Utiliza como representación las redes semánticas tanto para los ejemplos como para los conceptos
- Se centra en el dominio de aplicación del mundo de bloques

Un Quasi-Ejemplo es un ejemplo **NEGATIVO** que se diferencia en **UNA SOLA PROPIEDAD** con el **MODELO**

## Elementos del algoritmo:

- Un lenguaje de **representación: redes semánticas**
- Un mecanismo de **cotejamiento**: Método para análisis de diferencias
- Un proceso de **generalización**: para incorporar ejemplos positivos al modelo
- Un proceso de **especialización**: para rechazar ejemplos negativos (*quasi-ejemplos*) con el modelo

Elementos del algoritmo:

- Un lenguaje de **representación: redes semánticas**
- Un mecanismo de **cotejamiento**: Método para análisis de diferencias
- Un proceso de **generalización**: para incorporar ejemplos positivos al modelo
- Un proceso de **especialización**: para rechazar ejemplos negativos (*quasi-ejemplos*) con el modelo

Elementos del algoritmo:

- Un lenguaje de **representación: redes semánticas**
- Un mecanismo de **cotejamiento**: Método para análisis de diferencias
- Un proceso de **generalización**: para incorporar ejemplos positivos al modelo
- Un proceso de **especialización**: para rechazar ejemplos negativos (*quasi-ejemplos*) con el modelo

Elementos del algoritmo:

- Un lenguaje de **representación: redes semánticas**
- Un mecanismo de **cotejamiento**: Método para análisis de diferencias
- Un proceso de **generalización**: para incorporar ejemplos positivos al modelo
- Un proceso de **especialización**: para rechazar ejemplos negativos (*quasi-ejemplos*) con el modelo

## Descripción:

- Una red semántica o esquema de representación en Red es una forma de representación del conocimiento lingüístico
- Los conceptos y sus interrelaciones se representan mediante un grafo. En caso de que no existan ciclos, estas redes pueden ser visualizadas como árboles.
- En un grafo o red semántica los elementos semánticos se representan por nodos y las relaciones por aristas.
- Las redes semánticas son usadas, entre otras cosas, para representar mapas conceptuales y mentales.

## Descripción:

- Una red semántica o esquema de representación en Red es una forma de representación del conocimiento lingüístico
- Los conceptos y sus interrelaciones se representan mediante un grafo. En caso de que no existan ciclos, estas redes pueden ser visualizadas como árboles.
- En un grafo o red semántica los elementos semánticos se representan por nodos y las relaciones por aristas.
- Las redes semánticas son usadas, entre otras cosas, para representar mapas conceptuales y mentales.



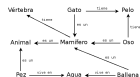
## Descripción:

- Una red semántica o esquema de representación en Red es una forma de representación del conocimiento lingüístico
- Los conceptos y sus interrelaciones se representan mediante un grafo. En caso de que no existan ciclos, estas redes pueden ser visualizadas como árboles.
- En un grafo o red semántica los elementos semánticos se representan por nodos y las relaciones por aristas.
- Las redes semánticas son usadas, entre otras cosas, para representar mapas conceptuales y mentales.

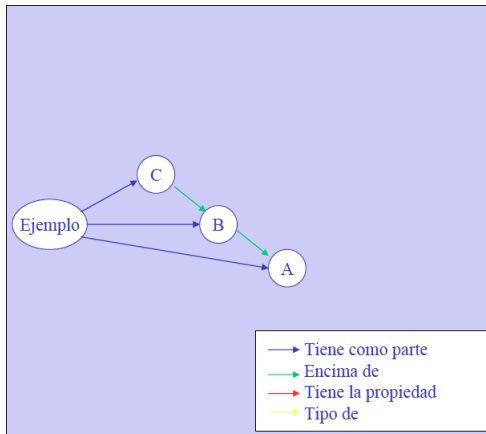
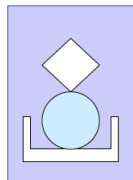
## Descripción:

- Una red semántica o esquema de representación en Red es una forma de representación del conocimiento lingüístico
- Los conceptos y sus interrelaciones se representan mediante un grafo. En caso de que no existan ciclos, estas redes pueden ser visualizadas como árboles.
- En un grafo o red semántica los elementos semánticos se representan por nodos y las relaciones por aristas.
- Las redes semánticas son usadas, entre otras cosas, para representar mapas conceptuales y mentales.

# Redes semánticas

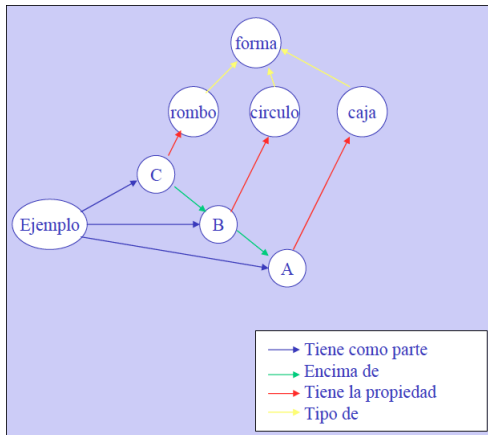
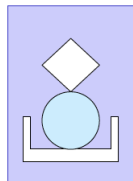


Ejemplo de construcción de una red semántica.

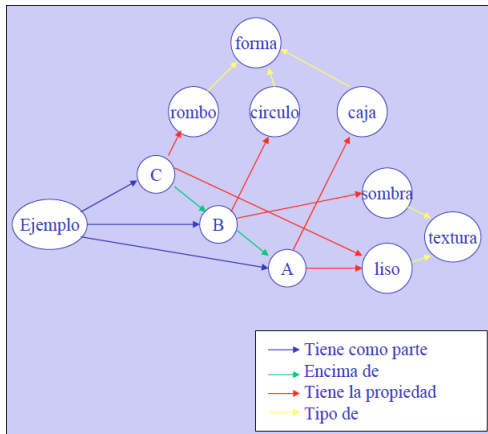
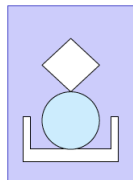


# Redes semánticas

## Ejemplo de construcción de una red semántica.

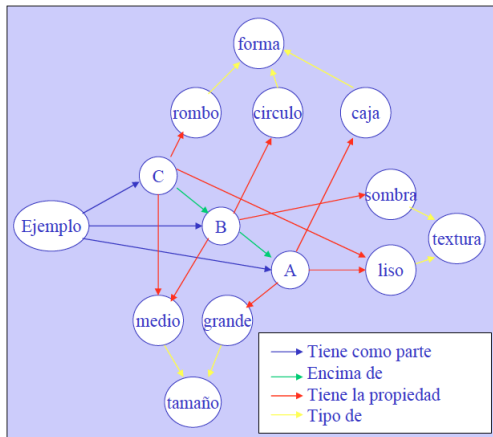
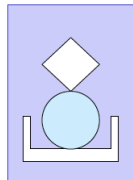


## Ejemplo de construcción de una red semántica.

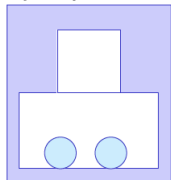


# Redes semánticas

## Ejemplo de construcción de una red semántica.

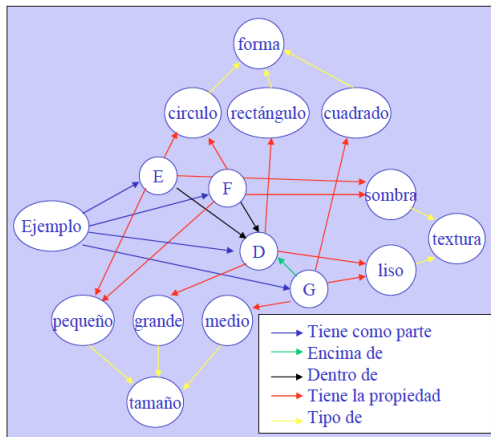
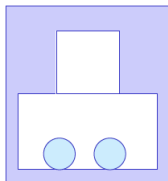


Ejemplo de construcción de una red semántica.





## Ejemplo 2



- Permite comparar dos redes semánticas.  
Corresponden a la definición actual del concepto y a la definición del nuevo ejemplo a tratar.
- Según sea el ejemplo positivo o negativo, se dirigirá la búsqueda a generalizar o especializar la definición.
- Sólo se pueden modificar las etiquetas de las relaciones entre nodos, nunca los nodos. (Aunque si agregar).
- Se introducen dos heurísticas, en la búsqueda, sobre las etiquetas: **require-link** y **forbid-link**.

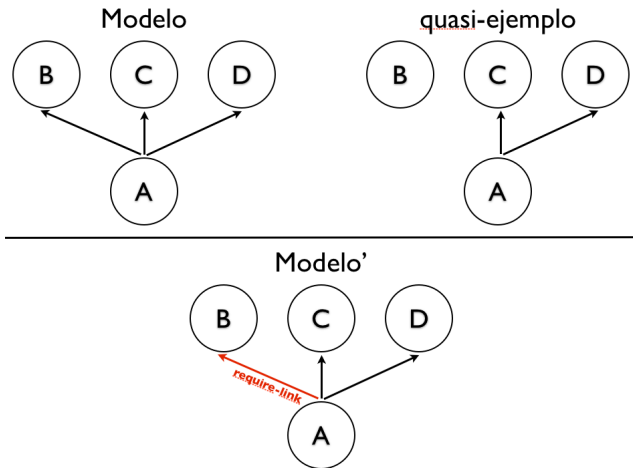
- Permite comparar dos redes semánticas.  
Corresponden a la definición actual del concepto y a la definición del nuevo ejemplo a tratar.
- Según sea el ejemplo positivo o negativo, se dirigirá la búsqueda a generalizar o especializar la definición.
- Sólo se pueden modificar las etiquetas de las relaciones entre nodos, nunca los nodos. (Aunque si agregar).
- Se introducen dos heurísticas, en la búsqueda, sobre las etiquetas: **require-link** y **forbid-link**.

- Permite comparar dos redes semánticas.  
Corresponden a la definición actual del concepto y a la definición del nuevo ejemplo a tratar.
- Según sea el ejemplo positivo o negativo, se dirigirá la búsqueda a generalizar o especializar la definición.
- Sólo se pueden modificar las etiquetas de las relaciones entre nodos, nunca los nodos. (Aunque si agregar).
- Se introducen dos heurísticas, en la búsqueda, sobre las etiquetas: **require-link** y **forbid-link**.

- Permite comparar dos redes semánticas.  
Corresponden a la definición actual del concepto y a la definición del nuevo ejemplo a tratar.
- Según sea el ejemplo positivo o negativo, se dirigirá la búsqueda a generalizar o especializar la definición.
- Sólo se pueden modificar las etiquetas de las relaciones entre nodos, nunca los nodos. (Aunque si agregar).
- Se introducen dos heurísticas, en la búsqueda, sobre las etiquetas: **require-link** y **forbid-link**.

- **Require-link:** es una heurística empleada cuando el modelo del concepto que está siendo aprendido (en evolución) tiene una etiqueta  $k$  en un lugar donde un quasi-ejemplo no. Entonces en la red semántica que representa el concepto esa etiqueta se transforma en **debe** (must).

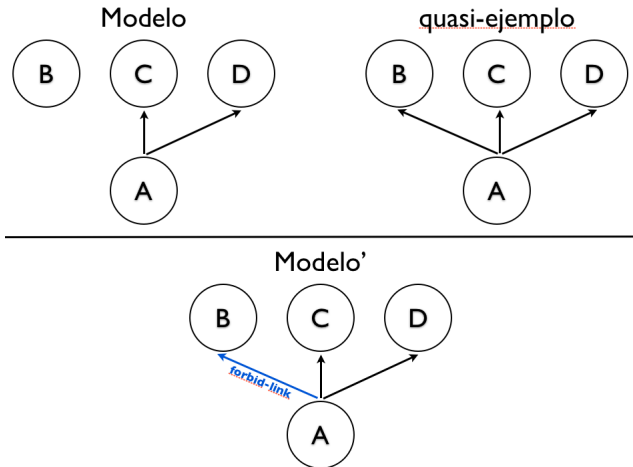
# Required-link



- **Forbidden-link**: esta heurística se aplica cuando un quasi-ejemplo tiene una etiqueta  $i$  en un lugar donde el modelo no. Entonces una etiqueta **no-debe** (must-not) se coloca en el modelo actual del concepto.



# Forbidden-link



Cotejar el ejemplo (positivo) con el modelo actual del concepto.

- Procesar todas las diferencias
  - Si falta una etiqueta, eliminarla del concepto
  - Si hay diferencia en el valor de una propiedad, modificar el rango de la propiedad
  - Si una etiqueta apunta a una clase diferente
    - Si la clase pertenece a una jerarquía, subir en la jerarquía
    - Si la clase no pertenece a una jerarquía, eliminarla

Cotejar el ejemplo (negativo) con el modelo actual del concepto

- Si hay más de una diferencia, ignorar el ejemplo
- Si hay una única diferencia:
  - Si el modelo tiene una etiqueta que el ejemplo no tiene, generar una etiqueta “required-link”
  - Si el ejemplo tiene una etiqueta que el concepto no tiene, generar una etiqueta “forbidden-link”

- I. Tomar como modelo inicial la descripción de la primera instancia positiva del concepto.  
Llamar a esta descripción la **definición del concepto**.
- II. Examinar la descripción de otras instancias positivas conocidas del concepto. **Generalizar** la definición del concepto para incluirlas.
- III. Examinar las descripciones de los *quasi-ejemplos* del concepto. **Especializar** la definición del concepto para excluirlas.
- IV. Los pasos 2 y 3 se van intercalando a medida que se van tratando ejemplos positivos y negativos del concepto.

# Propiedades del algoritmo

- Conservador: si existen dudas sobre lo que hay que aprender, mejor no aprender
- El aprendizaje se realiza en pasos pequeños (ley de Martin, no puedes aprender algo a menos que casi lo sepas de antemano)
- El algoritmo se ha descrito de manera no determinista ya que en cualquier punto puede haber varias posibles generalizaciones o especializaciones aplicables.
- La elección no necesariamente llevará a la hipótesis más sencilla; incluso puede ocurrir que lleguemos a una situación en que ninguna modificación sencilla de la hipótesis la haga consistente con todos los ejemplos. En ese punto será necesario un backtracking (vuelta atrás) al punto de elección

# Dificultades del algoritmo de Winston

- hay que comprobar que cada modificación es consistente con todos los ejemplos
- Es muy sensible al orden en el que se presentan los ejemplos
- es difícil encontrar una buena heurística para elegir la mejor modificación y el backtracking es muy costoso (algoritmo del espacio de versiones de Mitchell)
- Ya a finales de los 70 el algoritmo de Winston se aplicó en Meta-Dendral, un sistema para la predicción de resultados de espectrometría. Generó conocimiento que fue publicado en una revista de química analítica

- 1 Introducción
- 2 Análisis de diferencias
  - Winston
    - Redes semánticas
    - Método de cotejamiento
    - Generalización
    - Especialización
- 3 Espacio de versiones
  - Find-S
  - List-Then-Eliminate
  - Eliminación de candidatos
- 4 Bibliografía

- Propuesto por Mitchell en 1982
- Presenta un marco unificado para la adquisición de conceptos, independiente de la representación a utilizar
- El objetivo es el mismo que en el esquema anterior, es decir, producir una descripción de un concepto a partir de un entrenamiento con ejemplos positivos y negativos.
- No se ve afectado por el orden en que se presentan los ejemplos.
- en lugar de describir un único concepto este esquema mantiene un conjunto de descripciones posibles hasta arribar a la definición del mismo.



- **Hipótesis**: representación de un concepto
- **Espacio de hipótesis**: el conjunto de todos los conceptos que pueden ser descritos con la representación escogida
- **Espacio de versiones**: Conjunto de hipótesis coherentes con el conjunto de ejemplos positivos y negativos estudiado, es decir, que reconocen a todos los ejemplos positivos y excluyen a todos los ejemplos negativos.

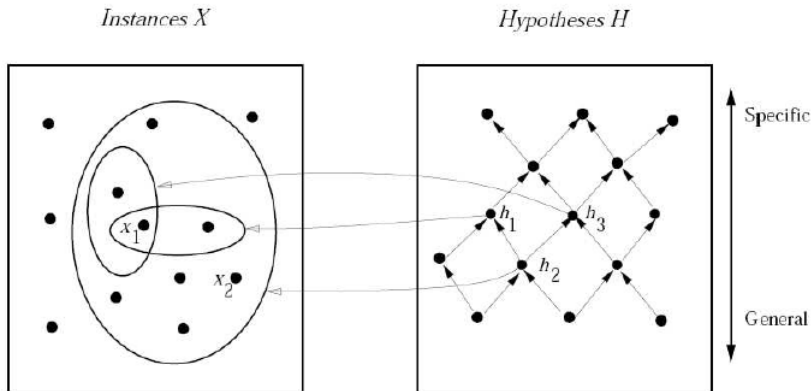
- **Hipótesis:** representación de un concepto
- **Espacio de hipótesis:** el conjunto de todos los conceptos que pueden ser descritos con la representación escogida
- **Espacio de versiones:** Conjunto de hipótesis coherentes con el conjunto de ejemplos positivos y negativos estudiado, es decir, que reconocen a todos los ejemplos positivos y excluyen a todos los ejemplos negativos.

- **Hipótesis**: representación de un concepto
- **Espacio de hipótesis**: el conjunto de todos los conceptos que pueden ser descritos con la representación escogida
- **Espacio de versiones**: Conjunto de hipótesis coherentes con el conjunto de ejemplos positivos y negativos estudiado, es decir, que reconocen a todos los ejemplos positivos y excluyen a todos los ejemplos negativos.

- **Hipótesis**: representación de un concepto
- **Espacio de hipótesis**: el conjunto de todos los conceptos que pueden ser descritos con la representación escogida
- **Espacio de versiones**: Conjunto de hipótesis coherentes con el conjunto de ejemplos positivos y negativos estudiado, es decir, que reconocen a todos los ejemplos positivos y excluyen a todos los ejemplos negativos.

# Espacio de versiones

- **Orden parcial:**  $h_1$  es más general que  $h_2$  ( $h_1 >_g h_2$ ) si el conjunto de instancias cubierto por  $h_2$  es un subconjunto del conjunto de instancias cubierto por  $h_1$



## ■ Representación del espacio de versiones:

- Enumeración de todas las hipótesis (inviable por su tamaño)
- Conjunto más general (G) y más específico (S)

- **Representación del espacio de versiones:**
  - Enumeración de todas las hipótesis (inviabile por su tamaño)
  - Conjunto más general (G) y más específico (S)

## ■ Representación del espacio de versiones:

- Enumeración de todas las hipótesis (inviabile por su tamaño)
- Conjunto más general (G) y más específico (S)



- $\geq_g$  no depende del concepto que se va a aprender
- el operador define *un orden parcial* sobre el conjunto de las hipótesis
- está también la versión estricta:  $>_g$
- .. más específica que:  $\leq_g$

- Comienza con la hipótesis más específica:  $(\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$
- Generaliza si el ejemplo positivo no está cubierto por la hipótesis

## Proceso de generalización

- Hipótesis Actual:  $(h_1, h_2, h_3, h_4, h_5)$
- Ejemplo nuevo:  $(e_1, e_2, e_3, e_4, e_5)$
- Hipótesis generalizada:  $(g_1, g_2, g_3, g_4, g_5)$ 
  - Si  $h_i = ?$  Entonces  $g_i = ?$
  - Si  $h_i = e_i$  Entonces  $g_i = h_i$
  - Si  $h_i \neq e_i$  Entonces  $g_i = ?$
  - Si  $h_i = \emptyset$  Entonces  $g_i = e_i$

Proceso de especialización:

- Hipótesis Actual:  $(h_1, h_2, h_3, h_4, h_5)$
- Ejemplo negativo:  $(e_1, e_2, e_3, e_4, e_5)$
- Hipótesis especializadas:  $[G_1, G_2, G_3, \dots]$ 
  - Si  $h_i = ?$  Entonces generar una hipótesis por cada  $g_i \neq e_i$
  - Si  $h_i \neq e_i$  y  $h_i \neq ?$  entonces la hipótesis no cubre el ejemplo negativo
  - Si  $h_i = e_i$  Entonces  $g_i = \emptyset$

$$\frac{\text{Valor}_1 \cdot \text{Valor}_2}{\emptyset}$$
$$\frac{\sum \text{Valor}_i}{\text{Total} - \text{Valor}_1}$$

# Algoritmo Find-S

DUAL FIND-S

↳ envía de especialista generalista.

la hipótesis cubre el ejemplo +.

Pseudocódigo:  $O(n \cdot m)$

```
1 Initialize the most specific hypothesis h _
2
3 For each positive instance x:
4   For each attribute  $a_i$  in h:
5     if  $x[a_i]$  satisfy the constraint of h: — Si el ato satisface la hipótesis
6       Nothing
7     else: — Si no la satisface.
8       Replace  $a_i$  in h with a generalization
9
10 return h
```

— generalista demasiado y puede cubrir algún ejemplo —

Función Especialista → Devuelve la lista de posibles hipótesis.

# Algoritmo Find-S

*Big data*

$x_1 = \langle \text{Sunny Warm Normal Strong Warm Same} \rangle, +$   
 $x_2 = \langle \text{Sunny Warm High Strong Warm Same} \rangle, +$   
 $x_3 = \langle \text{Rainy Cold High Strong Warm Change} \rangle, -$   
 $x_4 = \langle \text{Sunny Warm High Strong Cool Change} \rangle, +$

$h_0 = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$

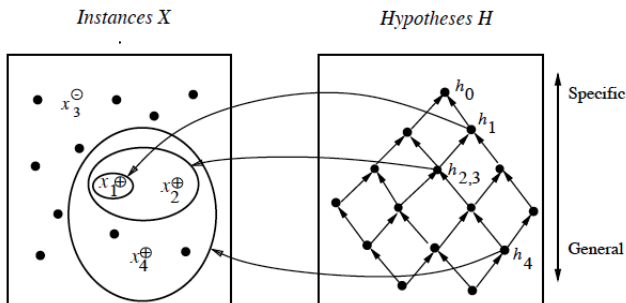
$h_1 = (\text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same})$

$h_2 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same})$

$h_3 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same})$

$h_4 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ?)$

*maximally specific*



# Algoritmo Find-S

$x_1 = \langle \text{Sunny Warm Normal Strong Warm Same} \rangle, +$   
 $x_2 = \langle \text{Sunny Warm High Strong Warm Same} \rangle, +$   
 $x_3 = \langle \text{Rainy Cold High Strong Warm Change} \rangle, -$   
 $x_4 = \langle \text{Sunny Warm High Strong Cool Change} \rangle, +$

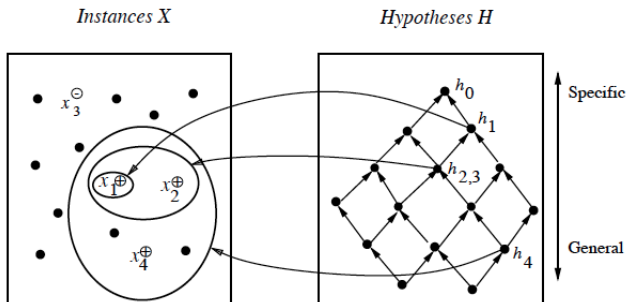
$h_0 = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$

$h_1 = (\text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same})$

$h_2 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same})$

$h_3 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same})$

$h_4 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ?)$



# Algoritmo Find-S

$x_1 = \langle \text{Sunny Warm Normal Strong Warm Same} \rangle, +$   
 $x_2 = \langle \text{Sunny Warm High Strong Warm Same} \rangle, +$   
 $x_3 = \langle \text{Rainy Cold High Strong Warm Change} \rangle, -$   
 $x_4 = \langle \text{Sunny Warm High Strong Cool Change} \rangle, +$

$h_0 = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$

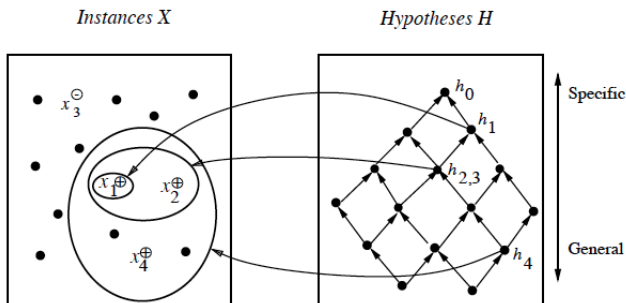
$h_1 = (\text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same})$

$h_2 = (\text{Sunny}, \text{Warm}, \text{?}, \text{Strong}, \text{Warm}, \text{Same})$

$h_3 = (\text{Sunny}, \text{Warm}, \text{?}, \text{Strong}, \text{Warm}, \text{Same})$

$h_4 = (\text{Sunny}, \text{Warm}, \text{?}, \text{Strong}, \text{?}, \text{?})$

Normal · High  
?





# Algoritmo Find-S

$x_1 = \langle \text{Sunny Warm Normal Strong Warm Same} \rangle, +$   
 $x_2 = \langle \text{Sunny Warm High Strong Warm Same} \rangle, +$   
 $x_3 = \langle \text{Rainy Cold High Strong Warm Change} \rangle, -$   
 $x_4 = \langle \text{Sunny Warm High Strong Cool Change} \rangle, +$

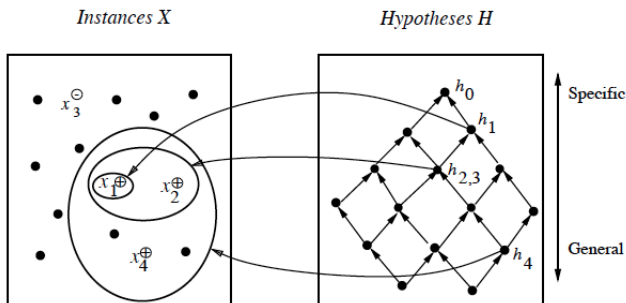
$h_0 = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$

$h_1 = (\text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same})$

$h_2 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same})$

$h_3 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same})$

$h_4 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ?)$



# Algoritmo Find-S

$x_1 = \langle \text{Sunny Warm Normal Strong Warm Same} \rangle, +$   
 $x_2 = \langle \text{Sunny Warm High Strong Warm Same} \rangle, +$   
 $x_3 = \langle \text{Rainy Cold High Strong Warm Change} \rangle, -$   
 $x_4 = \langle \text{Sunny Warm High Strong Cool Change} \rangle, +$

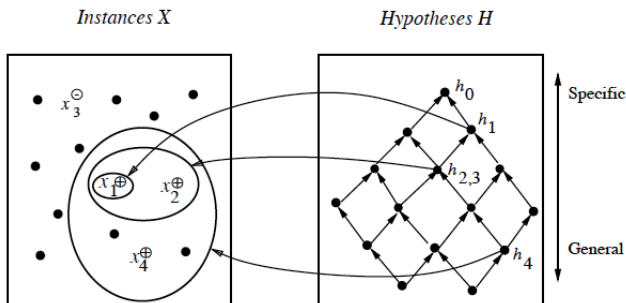
$h_0 = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$

$h_1 = (\text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same})$

$h_2 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same})$

$h_3 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same})$

$h_4 = (\text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ?)$



- Siempre se va a generar una hipótesis consistente con los ejemplos (se ignoran los negativos)
- No puede asegurar que se haya aprendido el concepto correcto, porque coge una de las hipótesis posibles.
- No soporta ruido en los ejemplos positivos

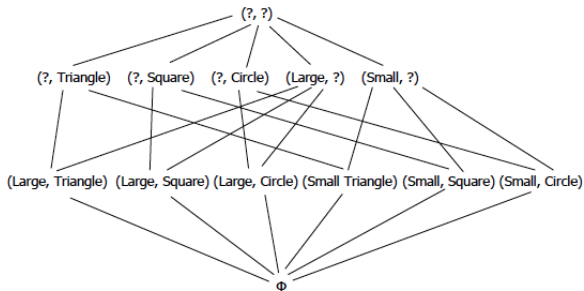
# Todas las hipótesis

- Construcción del espacio de hipótesis completo.
- Dado un ejemplo positivo, se excluyen las hipótesis que no lo cubren
- Dado un ejemplo negativo, se excluyen todas las hipótesis que si lo cubren

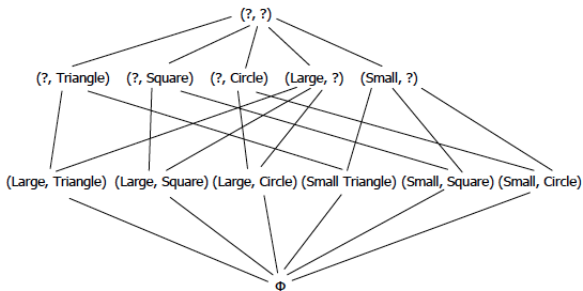
Por ejemplo:

Atributo	Valor
Size	Large, Small
Shape	Triangle , Square, Circle

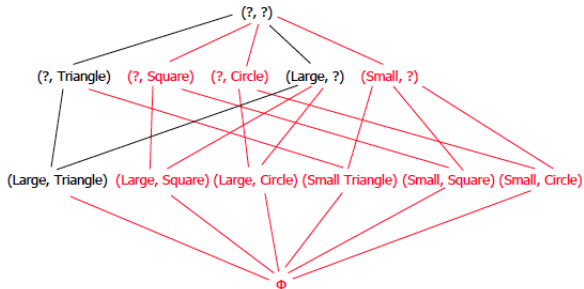
## Example 1



1<sup>st</sup> training example: (Large, Triangle)  $\rightarrow$  +

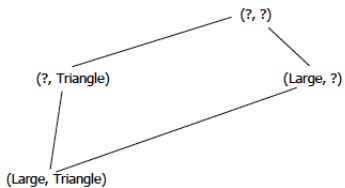


1<sup>st</sup> training example: (Large, Triangle)  $\rightarrow +$



Remove all concepts that do not include (Large, Triangle)

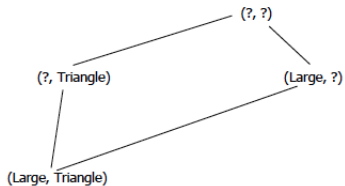
I.e., remove all concept descriptions that (Large, Triangle) does not match



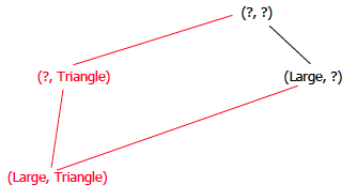
Updated version space after  
 $(\text{Large}, \text{Triangle}) \rightarrow +$



2<sup>nd</sup> training example: (Large, Circle)  $\rightarrow +$



2<sup>nd</sup> training example: (Large, Circle) → +



Remove all concept descriptions that (Large, Circle) does not match

(?, ?)

(Large, ?)

Updated version space after  
(Large, Triangle)  $\rightarrow$  +  
(Large, Circle)  $\rightarrow$  +

3<sup>rd</sup> training example: (Small, Circle) → -

(?, ?)

(Large, ?)

3<sup>rd</sup> training example: (Small, Circle) → -

(?, ?)

(Large, ?)

Remove all concept descriptions that (Small, Circle) does match

(Large, ?)

Updated version space after  
(Large, Triangle)  $\rightarrow +$   
(Large, Circle)  $\rightarrow +$   
(Small, Circle)  $\rightarrow -$

## Pseudocódigo

```
1 for each example x in dataset:  
2   for each hipotesis h in VS:  
3     if x not satisfy h:  
4       remove h  
5  
6 Return remaining VS
```

- Las opciones anteriores son demasiado costosas.
- Debemos de tener una opción que sea más tratable.
- El algoritmo de Eliminación de Candidatos mantiene “las cotas” de hipótesis superior e inferior de todas las hipótesis consistentes con los ejemplos.
- Cada tratamiento de ejemplos generaliza y especializa el grafo de las hipótesis, para obtener un espacio final compatible.



Algoritmo:

- Entrada: Conjunto de datos.
- Salida:
  - $G$  = Hipótesis genéricas maximales
  - $S$  = Hipótesis específicas maximales
- Las hipótesis se representan en un retículo con orden parcial

# Eliminación de Candidatos

## Pseudocódigo:

1. Sea  $G$  el conjunto de elementos de máxima generalidad de  $H$ .
2. Sea  $S$  el conjunto de elementos de máxima especificidad de  $H$ .

3. Para cada ejemplo  $d$  del conjunto de entrenamiento  $D$ :

3.1 Si  $d$  es un ejemplo positivo, entonces:

3.1.1 Eliminar de  $G$  cualquier hipótesis inconsistente con  $d$ .

3.1.2 Para cada hipótesis  $s$  de  $S$  inconsistente con  $d$ :

- \* Eliminar  $s$  de  $S$ .
- \* Incluir en  $S$  todas las generalizaciones minimales  $h$  de  $s$ , tales que  $h$  es consistente con  $d$  y existe una hipótesis en  $G$  más general que  $h$ .
- \* Eliminar de  $S$  aquellas hipótesis tales que exista en  $S$  otra hipótesis más general.

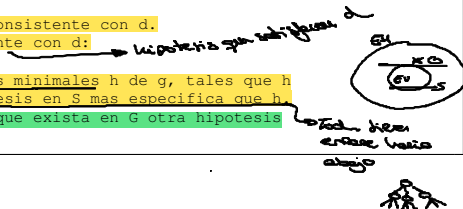
3.2 Si  $d$  es un ejemplo negativo, entonces:

3.2.1 Eliminar de  $S$  cualquier hipótesis inconsistente con  $d$ .

3.2.2 Para cada hipótesis  $g$  de  $G$  inconsistente con  $d$ :

- \* Eliminar  $g$  de  $G$ .
- \* Incluir en  $G$  todas las especializaciones minimales  $h$  de  $g$ , tales que  $h$  es consistente con  $d$  y existe una hipótesis en  $S$  más específica que  $h$ .
- \* Eliminar de  $G$  aquellas hipótesis tales que exista en  $G$  otra hipótesis más específica.

→ Eliminar Redundantes



1. Let  $G$  be the set of elements of maximum generality of  $H$ .
2. Let  $S$  be the set of elements of maximum specificity of  $H$ .
3. For each example  $d$  of training set  $D$ :
  - 3.1 If  $d$  is a positive example, then:
    - 3.1.1 Remove from  $G$  any hypothesis inconsistent with  $d$ .
    - 3.1.2 For each hypothesis  $s$  of  $S$  inconsistent with  $d$ :
      - \* Delete  $s$  from  $S$ .
      - \* Include in  $S$  all minimal generalizations  $h$  of  $s$ , such that  $h$  is consistent with  $d$  and there is a hypothesis in  $G$  more general than  $h$ .
      - \* Eliminate from  $S$  those hypotheses such that another hypothesis exists in  $S$  more general.
  - 3.2 If  $d$  is a negative example, then:
    - 3.2.1 Eliminate from  $S$  any hypotheses inconsistent with  $d$ .
    - 3.2.2 For each hypothesis  $g$  of  $G$  inconsistent with  $d$ :
      - \* Remove  $g$  from  $G$ .
      - \* Include in  $G$  all minimal specializations  $h$  of  $g$ , such that  $h$  is consistent with  $d$  and there is a hypothesis in  $S$  more specific than  $h$ .
      - \* Eliminate from  $G$  those hypotheses such that there is another hypothesis in  $G$  more specific.

# Ejemplo

Paso 0:  $S_0 = \{ \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle \}$ ,  $G_0 = \{ \langle ?, ?, ?, ?, ? \rangle \}$

# Ejemplo

Paso 1:

\* Traer Algoritmo Impreso

- Ejemplo **positivo**:  $\langle Sol; Templ; Normal; Fuerte; Templ; Igual \rangle$
- Nada que eliminar de  $G_0$
- Generalización minimal de  $S_0$ :  
 $\langle Sol; Templ; Normal; Fuerte; Templ; Igual \rangle$
- Esta generalización es más específica que la hipótesis de  $G_0$
- Luego:

- $G1 = \{ \langle ?; ?; ?; ?; ?; ? \rangle \}$

Caso de hipótesis

- $S1 = \{ \langle Sol; Templ; Normal; Fuerte; Templ; Igual \rangle \}$

## Paso 2:

- Ejemplo **positivo**:  $\langle Sol; Templ; Alta; Fuerte; Templ; Igual \rangle$
- Nada que eliminar de G1
- Generalización minimal de  
 $S1 : \langle Sol; Templ; ?; Fuerte; Templ; Igual \rangle$
- Esta generalización es más específica que la hipótesis de G1
- Luego:
  - $G2 = \{ \langle ?; ?; ?; ?; ?; ? \rangle \}$
  - $S2 = \{ \langle Sol; Templ; ?; Fuerte; Templ; Igual \rangle \}$

# Ejemplo

## Paso 3:

- Ejemplo **negativo**:  
 $\langle \text{Lluvia; Fria; Alta; Fuerte; Templada; Cambio} \rangle$
- Nada que eliminar de S2.
- Especializaciones minimales de G2 que son mas generales que la hipotesis de S2:  
 $\langle \text{Sol; ?; ?; ?; ?; ?} \rangle$ ,  $\langle ?; \text{Templ; ?; ?; ?; ?} \rangle$  y  $\langle ?; ?; ?; ?; ?; \text{Igual} \rangle$ .
- Luego:
  - $S3 = \{ \langle \text{Sol; Templ; ?; Fuerte; Templ; Igual} \rangle \}$
  - $G3 = \{ \langle \text{Sol; ?; ?; ?; ?; ?} \rangle$ ,  $\langle ?; \text{Templ; ?; ?; ?; ?} \rangle$  y  $\langle ?; ?; ?; ?; ?; \text{Igual} \rangle \}$

## Paso 4:

- Ejemplo **positivo**:  $\langle \text{Sol}; \text{Templ}; \text{Alta}; \text{Fuerte}; \text{Fria}; \text{Cambio} \rangle$
- Eliminamos de  $G3$  la hipótesis:  $\langle ?; ?; ?; ?; ?; \text{Igual} \rangle$
- Generalización minimal de  $S3$ :  $\langle \text{Sol}; \text{Templ}; ?; \text{Fuerte}; ?; ? \rangle$ .
- Luego:
  - $S4 = \{ \langle \text{Sol}; \text{Templ}; ?; \text{Fuerte}; ?; ? \rangle \}$
  - $G4 = \{ \langle \text{Sol}; ?; ?; ?; ?; ? \rangle, \langle ?; \text{Templ}; ?; ?; ?; ? \rangle \}$



Sean  $S$  y  $G$  obtenidos por eliminación de candidatos

- Si  $S$  y  $G$  son no vacíos, resultan ser respectivamente la cota específica y cota general del espacio de versiones (respecto del conjunto de entrenamiento)
- Si  $S = G = \{h\}$ , entonces  $h$  es la única hipótesis de  $H$  consistente con todos los ejemplos
- Si  $S = G = \emptyset$ , no existe  $h \in H$  consistente con los ejemplos

Convergencia hacia el concepto objetivo, siempre que:

- Conjunto de entrenamiento suficientemente grande
- Ejemplos sin errores (ausencia de ruido)
- El concepto objetivo esta en  $H$

- 1 Introducción
- 2 Análisis de diferencias
  - Winston
    - Redes semánticas
    - Método de cotejamiento
    - Generalización
    - Especialización
- 3 Espacio de versiones
  - Find-S
  - List-Then-Eliminate
  - Eliminación de candidatos
- 4 Bibliografía

- Mitchell (1997): “Machine Learning”. McGraw-Hill.
- A. Moreno Ribas y otros (1994). “Aprendizaje Automático.” Ediciones UPC (Universidad Politécnica de Cataluña)
- Método Winston