Tema 6: Aprendizaje de reglas

José A. Alonso Jiménez
Francisco Jesús Martín Mateos
José Luis Ruiz Reina

Dpto. de Ciencias de la Computación e Inteligencia Artificial

Universidad de Sevilla

Contenido

- Reglas proposicionales
- Ejemplo de aprendizaje de reglas proposicionales
- Algoritmos de cobertura
- Reglas de primer orden: programación lógica inductiva
- El algoritmo FOIL
  - Generación de especializaciones
  - Ganancia de información
  - Ejemplo

Reglas proposicionales

- **Reglas de clasificación:**
  - **R1: Si** $Cielo = Soleado \wedge Humedad = Alta$ **Entonces** $Jugar\_Tenis = No$
  - **R2: Si** $Astigmatismo = + \wedge Lagrima = Normal$ **Entonces** $Lente = Rigida$
- **Ventaja de usar el formalismo de reglas**
  - **Claridad**
  - **Modularidad**
  - **Expresividad: pueden representar cualquier conjunto de instancias**
  - **Métodos generalizables a primer orden de manera natural**
  - **Formalismo usado en S.B.C.**
- **Reglas y árboles de decisión**
  - **Fácil traducción de árbol a reglas**
  - **No tan fácil a la inversa**

Aprendizaje de reglas

- **Objetivo: aprender un conjunto de reglas consistente con los ejemplos**
  - **Una regla *cubre* un ejemplo si el ejemplo satisface las condiciones**
  - **Lo cubre *correctamente* si además el valor del atributo en la conclusión de la regla coincide con el valor que el ejemplo toma en ese atributo**
  - **De la tabla siguiente, R2 cubre E_4, E_8, E_{12}, E_{16}, E_{20} y E_{24}, de los cuales cubre correctamente E_4, E_8, E_{12} y E_{20}.**
- **Una medida del ajuste de una regla R a un conjunto de ejemplos D:**
  - ***Frecuencia relativa*: p/t (donde t = ejemplos cubiertos por R en D, p = ejemplos correctamente cubiertos). Notación: $FR(R, D)$**
- **Algoritmos de aprendizaje de reglas:**
  - **ID3 + traducción a reglas**
  - **Cobertura**
  - **Algoritmos genéticos**
  - **...**

Un conjunto de entrenamiento

Ej.	Edad	Dignostico	Astigmatismo	Lagrima	Lente
E_1	Joven	Miope	-	Reducida	Ninguna
E_2	Joven	Miope	-	Normal	Blanda
E_3	Joven	Miope	+	Reducida	Ninguna
E_4	Joven	Miope	+	Normal	Rígida
E_5	Joven	Hipermétrope	-	Reducida	Ninguna
E_6	Joven	Hipermétrope	-	Normal	Blanda
E_7	Joven	Hipermétrope	+	Reducida	Ninguna
E_8	Joven	Hipermétrope	+	Normal	Rígida
E_9	Pre-presbicia	Miope	-	Reducida	Ninguna
E_{10}	Pre-presbicia	Miope	-	Normal	Blanda
E_{11}	Pre-presbicia	Miope	+	Reducida	Ninguna
E_{12}	Pre-presbicia	Miope	+	Normal	Rígida
E_{13}	Pre-presbicia	Hipermétrope	-	Reducida	Ninguna
E_{14}	Pre-presbicia	Hipermétrope	-	Normal	Blanda
E_{15}	Pre-presbicia	Hipermétrope	+	Reducida	Ninguna
E_{16}	Pre-presbicia	Hipermétrope	+	Normal	Ninguna
E_{17}	Presbicia	Miope	-	Reducida	Ninguna
E_{18}	Presbicia	Miope	-	Normal	Ninguna
E_{19}	Presbicia	Miope	+	Reducida	Ninguna
E_{20}	Presbicia	Miope	+	Normal	Rígida
E_{21}	Presbicia	Hipermétrope	-	Reducida	Ninguna
E_{22}	Presbicia	Hipermétrope	-	Normal	Blanda
E_{23}	Presbicia	Hipermétrope	+	Reducida	Ninguna
E_{24}	Presbicia	Hipermétrope	+	Normal	Ninguna

Aprendiendo reglas que cubren ejemplos

- **Aprender una regla para clasificar** $Lente = Rigida$
  - **Si** ? **Entonces** $Lente = Rigida$
  - **Alternativas para ? (y frecuencia relativa que tendría la regla resultante)**

```
Edad=Joven                    2/8
Edad=Pre-presbicia            1/8
Edad=Presbicia                1/8
Diagnostico=Miopía            3/12
Diagnostico=Hipermetropía     1/12
Astigmatismo=-                0/12
Astigmatismo=+                4/12  --> Mejor opción
Lágrima=Reducida              0/12
Lágrima=Normal                4/12  --> Mejor opción
```

- **Regla parcialmente aprendida**
  - **R1: Si** $Astigmatismo = +$ **Entonces** $Lente = Rigida$**, con** $FR(R1, D) = 4/12$

Aprendiendo reglas que cubren ejemplos

- **Continuamos para excluir ejemplos cubiertos incorrectamente**
  - **Ejemplos cubiertos por R1,** $D' = \{E_3, E_4, E_7, E_8, E_{11}, E_{12}, E_{15}, E_{16}, E_{19}, E_{20}, E_{23}, E_{24}\}$
  - **Si** $Astigmatismo = + \quad \wedge \quad ?$ **Entonces** $Lente = Rigida$
  - **Alternativas para ?**

```
Edad=Joven                    2/4
Edad=Pre-presbicia            1/4
Edad=Presbicia                1/4
Diagnostico=Miopía            3/6
Diagnostico=Hipermetropía     1/6
Lágrima=Reducida              0/6
Lágrima=Normal                4/6   --> Mejor opción
```

- **Regla parcialmente aprendida**
  - **R2: Si** $Astigmatismo = + \wedge Lagrima = Normal$ **Entonces** $Lente = Rigida$**, con** $FR(R2, D') = 4/6$

Aprendiendo reglas que cubren ejemplos

- **Continuamos para seguir excluyendo ejemplos cubiertos incorrectamente**
  - **Ejemplos cubiertos por R2,** $D'' = \{E_4, E_8, E_{12}, E_{16}, E_{20}, E_{24}\}$
  - **Si** $Astigmatismo = + \wedge Lagrima = Normal \quad \wedge \quad ? \quad$ **Entonces** $Lente = Rigida$
  - **Alternativas para ?**

```
Edad=Joven                    2/2   --> Mejor opción
Edad=Pre-presbicia            1/2
Edad=Presbicia                1/2
Diagnostico=Miopía            3/3   --> Mejor opción
Diagnostico=Hipermetropía     1/3
```

- **Regla finalmente aprendida (no cubre incorrectamente ningún ejemplo)**
  - **R: Si** $Astigmatismo = + \wedge Lagrima = Normal \wedge Diagnostico = Miopia$
    Entonces $Lente = Rigida$**, con** $FR(R, D'') = 3/3$

Aprendiendo reglas que cubren ejemplos

- **Quedan un ejemplo con** $Lente = Rigida$ **no cubierto por por** $R3$
  - **Comenzamos otra vez con " Si** $?$ **Entonces** $Lente = Rigida$**", pero ahora con** $D' = D \setminus \{E_4, E_{12}, E_{20}\}$
- **Reglas finalmente aprendidas para** $Lente = Rigida$**:**
  - **R: Si** $Astigmatismo = + \wedge Lagrima = Normal \wedge Diagnostico = Miopia$
    Entonces $Lente = Rigida$
  - **R': Si** $Edad = Joven \wedge Astigmatismo = + \wedge Lagrima = Normal$
    Entonces $Lente = Rigida$
  - **Nótese que cubren correctamente los 4 ejemplos de** $Lente = Rigida$ **(y se solapan)**
- **Ahora se podría continuar para aprender reglas que clasifiquen:**
  - $Lente = Blanda$
  - $Lente = Ninguna$

Aprendizaje de reglas por cobertura

- **Algoritmo para aprender un conjunto de reglas (a partir de D)**
  - **Reglas para predecir situaciones en las que un Atributo dado toma un valor v**
- **Aprendizaje-por-Cobertura(D,Atributo,v)**

```
1. Hacer Reglas-aprendidas igual a vacío
2. Hacer E igual a D
3. Mientras E contenga ejemplos cuyo valor de Atributo es v, hacer:
   3.1 Crear una regla R sin condiciones y conclusión Atributo=v
   3.2 Mientras que haya en E ejemplos cubiertos por R incorrectamente
        o no queden atributos que usar, hacer:
       3.2.1 Elegir la MEJOR condición A=w para añadir a R, donde A es
             un atributo que no aparece en R y w es un valor de los
             posibles que puede tomar A
       3.2.2 Actualizar R añadiendo la condición A=w a R
   3.3 Incluir R en  Reglas-aprendidas
   3.4 Actualizar E quitando los ejemplos cubiertos por R
4. Devolver Reglas-Aprendidas
```

Control en el algoritmo de cobertura

- **Bucle externo:**
  - **Añade reglas (en cada vuelta, la hipótesis se *generaliza*)**
  - **Cada regla añadida cubre algunos ejemplos, y todos ellos correctamente**
  - **Elimina en cada vuelta los ejemplos cubiertos por la regla añadida**
  - **Y se añaden reglas mientras queden ejemplos sin cubrir**
- **Bucle interno:**
  - **Añade condiciones a la regla (en cada vuelta, la regla se *especializa*)**
  - **Cada nueva condición excluye algunos ejemplos cubiertos incorrectamente**
  - **Y esto se hace mientras haya ejemplos que la regla cubre incorrectamente**
- **Cobertura frente a ID3**
  - **Aprende una regla cada vez, ID3 lo hace simultáneamente**
  - **ID3: elecciones de atributos**
  - **Cobertura: elcciones de parejas atributo-valor**

Algoritmo de cobertura (propiedades)

- **Diferentes criterios para elegir la mejor condición en cada vuelta del bucle interno:**
  - **Se añade la condición que produzca la regla con *mayor frecuencia relativa* (como en el ejemplo)**
  - **Se añade la que produzca *mayor ganancia de información*:**

$$p \cdot (log_2 \frac{p'}{t'} - log_2 \frac{p}{t})$$

    donde p'/t' es la frecuencia relativa *después* de añadir la condición y p/t es la frecuencia relativa *antes* de añadir la condición

- **Las reglas aprendidas por el algoritmo de cobertura se ajustan *perfectamente* al conjunto de entrenamiento (*peligro de sobreajuste*)**
  - **Podado de las reglas *a posteriori***
  - **Eliminar progresivamente condiciones hasta que no se produzca *mejora***
  - **Criterio probabilístico para decidir la mejora**

Insuficiencia expresiva de las reglas proposicionales

- **Ejemplo:**
  - **aprender el concepto "ser hija de" a partir de una base de datos de relaciones entre miembros de una familia, con ejemplos positivos y negativos del concepto**
  - **Dificultad de ser expresado en el modelo proposicional de pares atributo-valor**
- **Expresión del ejemplo en lógica de primer orden (familia 1)**
  - **Ejemplos positivos y negativos de la relación que se quiere aprender**

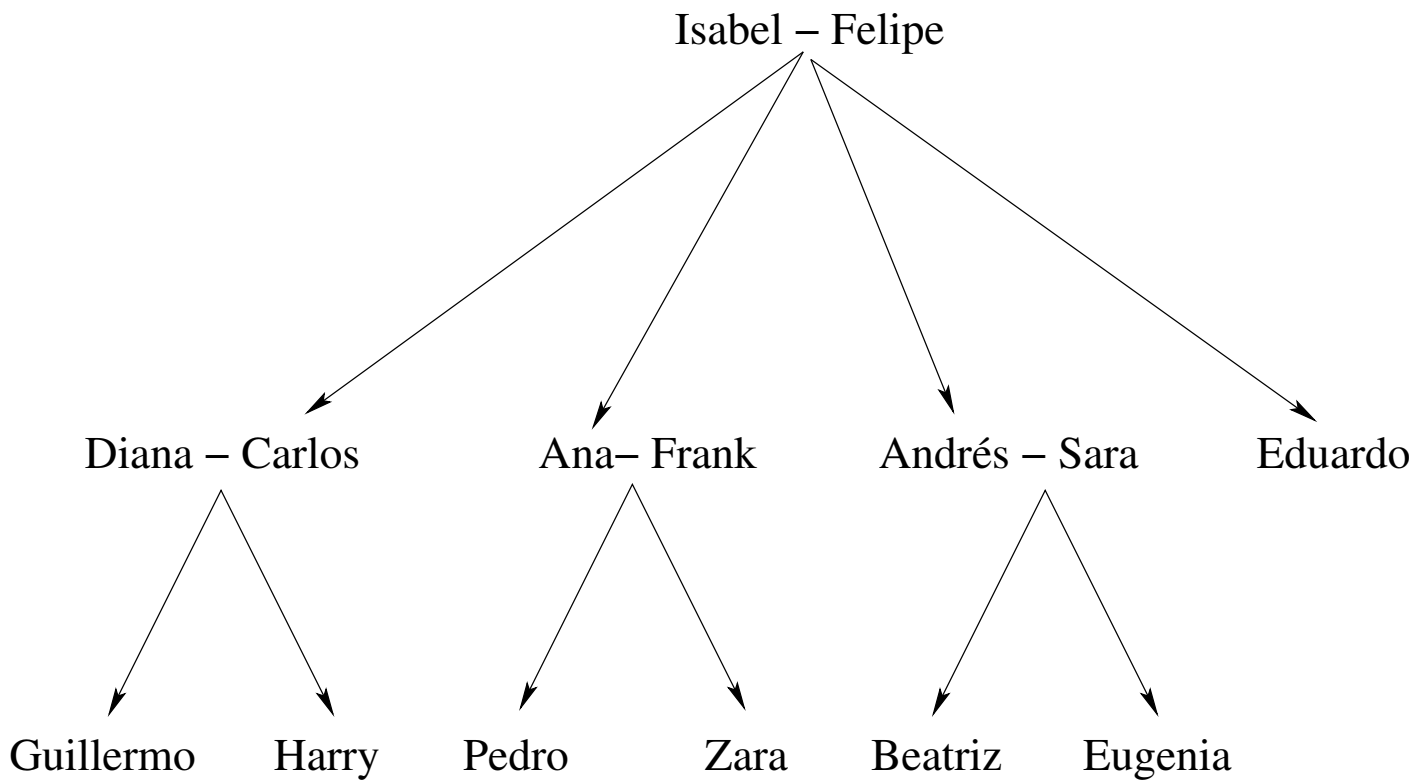
```
hija(maria,ana).          no(hija(tomas,ana)).
hija(eva,tomas).          no(hija(eva,ana)).
                          no(hija(eva,ignacio)).
```

  - **Conocimiento base**

```
hombre(tomas). hombre(ignacio). mujer(ana). mujer(eva). mujer(maria).
progenitor(ana,maria). progenitor(ana,tomas). progenitor(tomas,eva).
progenitor(tomas,ignacio).
```

  - **Regla aprendida:** `hija(A, B) :- progenitor(B, A),mujer(A)`

Ejemplo (familia 2)



Ejemplo (familia 2)

- **Aprender la relación "ser abuelo de"**
- **Ejemplos positivos (los restantes, negativos, CWA):**

```
abuelo(felipe,guillermo).   abuelo(felipe,harry).    abuelo(felipe,pedro).
abuelo(felipe,zara).        abuelo(felipe,beatriz).  abuelo(felipe,eugenia).
```

- **Conocimiento base:**

```
padre(felipe,carlos).  padre(felipe,ana). padre(felipe,andres).
padre(felipe,eduardo). padre(carlos,guillermo). padre(carlos,harry).
padre(mark,pedro). padre(mark,zara). padre(andres,beatriz). padre(andres,eugenia).

madre(isabel,carlos). madre(isabel,ana). madre(isabel,andres).
madre(isabel,eduardo). madre(diana,guillermo). madre(diana,harry).
madre(ana,pedro). madre(ana,zara). madre(sara,beatriz). madre(sara,eugenia).

progenitor(X,Y) :- padre(X,Y).
progenitor(X,Y) :- madre(X,Y).
```

- **Regla aprendida:** `abuelo(A, B) :- padre(A, C), progenitor(C, B)`

Programación Lógica Inductiva

- **Datos:**
  - **Ejemplos positivos:** $\boldsymbol{E}^{\oplus}$
  - **Ejemplos negativos:** $\boldsymbol{E}^{\ominus}$
  - **Conocimiento base:** $\boldsymbol{T}$
  - **Lenguaje de hipótesis:** $\boldsymbol{L}$
- **Condiciones:**
  - ***Necesidad a priori:*** $(\exists e^{\oplus} \in \boldsymbol{E}^{\oplus})[\boldsymbol{T} \nvdash e^{\oplus}]$
  - ***Consistencia a priori:*** $(\forall e^{\ominus} \in \boldsymbol{E}^{\ominus})[\boldsymbol{T} \nvdash e^{\ominus}]$
- **Objetivo:**
  - **Encontrar un conjunto finito** $\boldsymbol{H} \subset \boldsymbol{L}$ **tal que se cumplan**
    - ***Suficiencia a posteriori:*** $(\forall e^{\oplus} \in \boldsymbol{E}^{\oplus})[\boldsymbol{T} \bigcup \boldsymbol{H} \vdash e^{\oplus}]$
    - ***Consistencia a posteriori:*** $(\forall e^{\ominus} \in \boldsymbol{E}^{\ominus})[\boldsymbol{T} \bigcup \boldsymbol{H} \nvdash e^{\ominus}]$

Terminología en PLI

- Se aprenden *relaciones* (o *predicados*)
  - En lugar de pares atributo-valor
- Los *ejemplos* vienen dados por tuplas de constantes sobre los que la relación es cierta o falsa
  - En el ejemplo 1, `(eva,tomas)` es un *ejemplo positivo* de la relación abuelo y `(tomas,ana)` un *ejemplo negativo*
- Cobertura de un ejemplo mediante una regla:
  - Una regla *cubre* un ejemplo si el ejemplo satisface su cuerpo
  - En el ejemplo 1, la regla `hija(A, B) :- progenitor(B, A),mujer(A)` cubre el ejemplo `(maria,ana)` y no cubre el ejemplo `(eva,ignacio)`
  - Nótese que una regla *cubre correctamente* un ejemplo si y sólo si éste es positivo
- Objetivo:
  - Encontrar un conjunto de reglas PROLOG que cubra todos los ejemplos positivos y ninguno negativo

El algoritmo FOIL

- **FOIL(Ejemplos,Conocimiento-base,Predicado-objetivo)**

```
1. Hacer Reglas-aprendidas igual a vacío
2. Hacer E igual a Ejemplos
3. Mientras E contenga ejemplos positivos, hacer:
   3.1 Crear una regla R sin cuerpo y con cabeza P(X1,...,Xn)
       (donde P es el Predicado-objetivo y n su aridad)
   3.2 Mientras que haya en E ejemplos negativos cubiertos por R, hacer:
       3.2.1 GENERAR todos los posibles literales que pueden ser
             añadidos al cuerpo de la regla R
       3.2.2 De todos ellos, sea L el MEJOR
       3.2.2 Actualizar R añadiendo el literal L al cuerpo de R
   3.3 Incluir R en  Reglas-aprendidas
   3.4 Actualizar E quitando los ejemplos cubiertos por R
4. Devolver Reglas-Aprendidas
```

- **Debemos precisar `GENERAR` y `MEJOR`**

Generación de nuevos literales en FOIL

- **Literales tomados de los predicados del conocimiento base:**
  - **De la forma $Q(x_1, \ldots, x_n)$ donde Q es un predicado del conocimiento base y x_i son variables de la regla que se quiere extender o nuevas (al menos una de las variables ha de estar en la regla)**
  - **En el ejemplo 1, para ampliar la regla** `hija(A,B) :-`

```
hombre(A), hombre(B), mujer(A), mujer(B),
progenitor(C, A), progenitor(A,C), progenitor(C, B), progenitor(B, C),
progenitor(A, A), progenitor(B, A), progenitor(A, B), progenitor(B, B)
```

- **Literales de igualdad entre las variables que aparecen en la regla**
- **Negaciones de los anteriores**

Elección del mejor literal

- **El criterio más usado:**
  - **Dada una regla R y un literal L medimos la mejora producida en la cobertura al añadir un nuevo literal L a R para obtener una nueva regla R'**
  - **Se elige el que produzca *mayor ganancia de información*:**

$$t \cdot (log_2 \frac{p'}{p'+n'} - log_2 \frac{p}{p+n})$$

    donde R cubre p ejemplos positivos y n negativos, R' cubre p' ejemplos positivos y n' negativos y t es el número de ejemplos positivos cubiertos por R que siguen cubriéndose en R'

- **Otros criterios posibles**
  - **Frecuencia relativa**
  - **Contenido de información**

Literales que introducen nuevas variables

- **Ejemplos cubiertos en familia 1 por la regla** `hija(A,B) :-`
  - **Positivos:** `(maria,ana)` **y** `(eva,tomas)`
  - **Negativos:** `(tomas,ana)`, `(eva,ana)` **y** `(eva,ignacio)`
- **Ejemplos (extendidos) cubiertos por la regla** `hija(A,B) :- progenitor(B,C)`
  - **Pos.:** `(maria,ana,maria)`, `(maria,ana,tomas)`, `(eva,tomas,eva)` **y** `(eva,tomas,ignacio)`
  - **Neg.:** `(tomas,ana,maria)`, `(tomas,ana,tomas)`, `(eva,ana,maria)` **y** `(eva,ana,tomas)`
- **Al introducir nuevas variables:**
  - **Las tuplas *se extienden* con las nuevas ligaduras**
  - **Por tanto, el número de ejemplos cubiertos puede aumentar (tanto positivos como negativos)**
  - **Precisión del valor t en la fórmula de ganancia de información: un ejemplo positivo se considera que sigue cubierto al extender la regla si existe alguna extensión del ejemplo que queda cubierto por la regla ($t = 2$ en el caso anterior)**

Ejemplo familia 1 en FOIL

- **Parámetros:**

```
foil_predicates([hija/1,hombre/1,mujer/1,progenitor/2]).
foil_cwa(false).                    % No usa la hipótesis del mundo cerrado
foil_use_negations(false).          % No usa información negativa
foil_det_lit_bound(0).              % No añade literales determinados
```

- **Ejemplos y conocimiento base, dados anteriormente**

- **Sesión:**

```
?- foil(hija/2).
Uncovered positives:
[hija(maria, ana), hija(eva, tomas)]
Adding a clause ...
Specializing current clause:
hija(A, B).
Covered negatives:
[hija(tomas, ana), hija(eva, ana), hija(eva, ignacio)]
Covered positives:
[hija(maria, ana), hija(eva, tomas)]
```

Ejemplo familia 1 en FOIL

- **Sesión (continuación):**

```
Ganancia:  0.000 Cláusula: hija(A, B):-hombre(A)
Ganancia:  0.322 Cláusula: hija(A, B):-hombre(B)
Ganancia:  0.644 Cláusula: hija(A, B):-mujer(A)
Ganancia: -0.263 Cláusula: hija(A, B):-mujer(B)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(C, A)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(A, C)
Ganancia:  0.322 Cláusula: hija(A, B):-progenitor(C, B)
Ganancia:  0.644 Cláusula: hija(A, B):-progenitor(B, C)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(A, A)
Ganancia:  1.474 Cláusula: hija(A, B):-progenitor(B, A)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(A, B)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(B, B)
Specializing current clause:
hija(A, B) :- progenitor(B, A).

Covered negatives:
[hija(tomas, ana)]
Covered positives:
[hija(maria, ana), hija(eva, tomas)]
```

Ejemplo familia 1 en FOIL

- **Sesión (continuación):**

```
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(B, A), hombre(A)
Ganancia:  0.585 Cláusula: hija(A, B):-progenitor(B, A), hombre(B)
Ganancia:  1.170 Cláusula: hija(A, B):-progenitor(B, A), mujer(A)
Ganancia: -0.415 Cláusula: hija(A, B):-progenitor(B, A), mujer(B)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(B, A), progenitor(C, A)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(B, A), progenitor(A, C)
Ganancia:  0.585 Cláusula: hija(A, B):-progenitor(B, A), progenitor(C, B)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(B, A), progenitor(B, C)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(B, A), progenitor(A, A)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(B, A), progenitor(B, A)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(B, A), progenitor(A, B)
Ganancia:  0.000 Cláusula: hija(A, B):-progenitor(B, A), progenitor(B, B)

Clause found:
hija(A, B) :- progenitor(B, A), mujer(A).

Found definition:
hija(A, B) :- progenitor(B, A), mujer(A).
```

Ejemplo familia 2 en FOIL

- **Parámetros:**

```
foil_predicates([padre/2, madre/2, abuelo/2, progenitor/2]).
foil_cwa(true).                    % Usa la hipótesis del mundo cerrado
foil_use_negations(false).         % No usa información negativa
foil_det_lit_bound(0).             % No añade literales determinados
```

- **Ejemplos y conocimiento base, dados anteriormente**

- **Sesión:**

```
?- foil(abuelo/2).
Uncovered positives:
[abuelo(felipe, guillermo), abuelo(felipe, harry), abuelo(felipe, pedro),
 abuelo(felipe, zara), abuelo(felipe, beatriz), abuelo(felipe, eugenia)]
Adding a clause ...
Specializing current clause:
abuelo(A, B).
Covered negatives:
[abuelo(ana, ana), abuelo(ana, andres), ...]
Covered positives:
[abuelo(felipe, guillermo), abuelo(felipe, harry), abuelo(felipe, pedro),
 abuelo(felipe, zara), abuelo(felipe, beatriz), abuelo(felipe, eugenia)]
```

Ejemplo familia 2 en FOIL

- **Sesión (continuación):**

```
Ganancia:  0.000 Cláusula: abuelo(A, B):-padre(C, A)
Ganancia: 15.510 Cláusula: abuelo(A, B):-padre(A, C)
Ganancia:  3.510 Cláusula: abuelo(A, B):-padre(C, B)
.....
Ganancia:  0.000 Cláusula: abuelo(A, B):-madre(B, B)
Ganancia:  0.000 Cláusula: abuelo(A, B):-progenitor(C, A)
Ganancia:  9.510 Cláusula: abuelo(A, B):-progenitor(A, C)
Ganancia:  3.510 Cláusula: abuelo(A, B):-progenitor(C, B)
Ganancia:  0.000 Cláusula: abuelo(A, B):-progenitor(B, C)
Ganancia:  0.000 Cláusula: abuelo(A, B):-progenitor(A, A)
Ganancia:  0.000 Cláusula: abuelo(A, B):-progenitor(B, A)
Ganancia:  0.000 Cláusula: abuelo(A, B):-progenitor(A, B)
Ganancia:  0.000 Cláusula: abuelo(A, B):-progenitor(B, B)
Specializing current clause:
abuelo(A, B) :-padre(A, C).
Covered negatives:
 [abuelo(andres, ana), abuelo(andres, andres), ...]
Covered positives:
 [abuelo(felipe, guillermo), abuelo(felipe, harry), abuelo(felipe, pedro),
  abuelo(felipe, zara), abuelo(felipe, beatriz), abuelo(felipe, eugenia)]
```

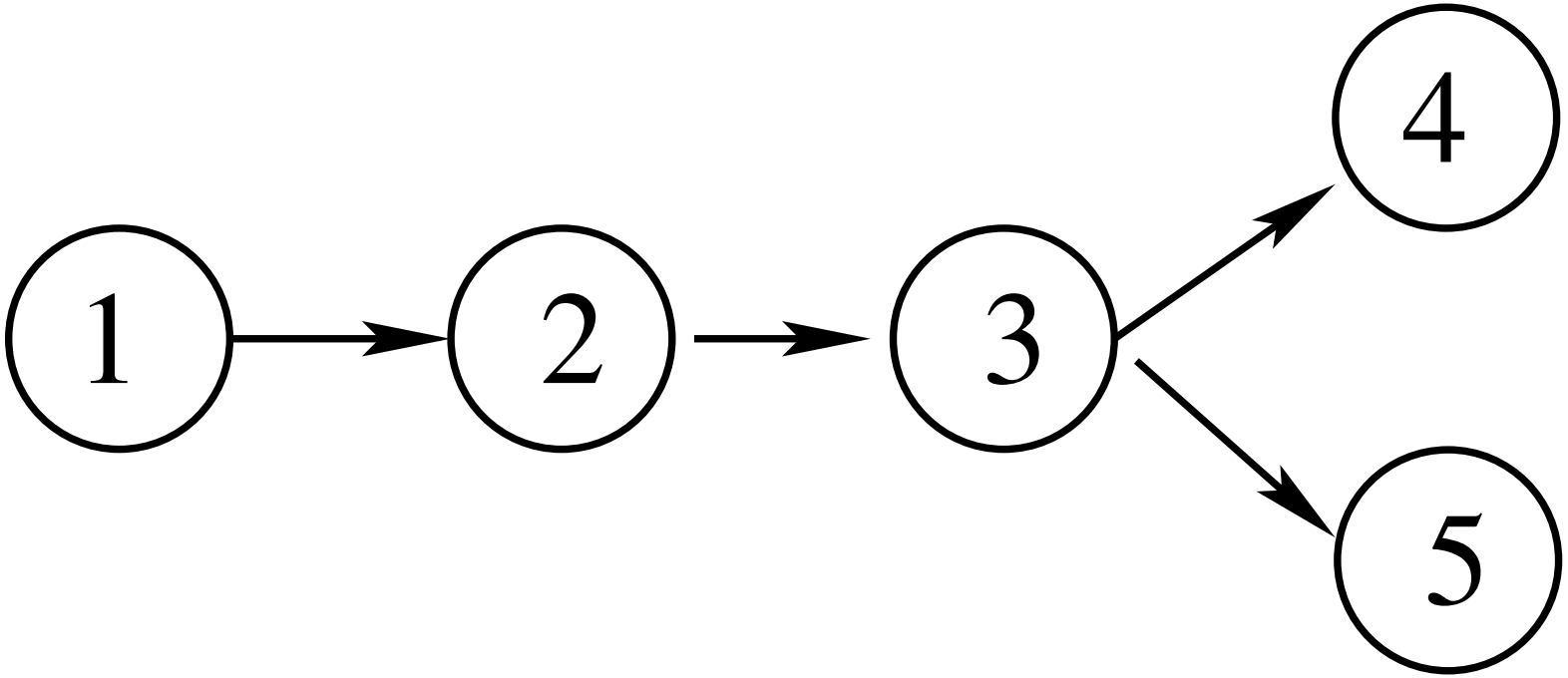
Ejemplo familia 2 en FOIL

- **Sesión (continuación):**

```
Ganancia:  0.000 Cláusula: abuelo(A, B):-padre(A, C), padre(D, A)
.....
Ganancia:  0.000 Cláusula: abuelo(A, B):-padre(A, C), padre(C, C)
Ganancia:  0.000 Cláusula: abuelo(A, B):-padre(A, C), madre(D, A)
.....
Ganancia:  0.000 Cláusula: abuelo(A, B):-padre(A, C), madre(C, C)
Ganancia:  0.000 Cláusula: abuelo(A, B):-padre(A, C), abuelo(C, A)
....
Ganancia:  0.000 Cláusula: abuelo(A, B):-padre(A, C), abuelo(C, C)
Ganancia:  0.000 Cláusula: abuelo(A, B):-padre(A, C), progenitor(D, A)
.....
Ganancia: 15.863 Cláusula: abuelo(A, B):-padre(A, C), progenitor(C, B)
....
Clause found:
 abuelo(A, B) :- padre(A, C), progenitor(C, B).
Found definition:
 abuelo(A, B) :-padre(A, C), progenitor(C, B).
```

Ejemplo con FOIL de relación recursiva

- Grafo



Ejemplo con FOIL de relación recursiva

- **Parámetros**

```
foil_predicates([camino/2, enlace/2]).
foil_cwa(true).                % Usa la hipótesis del mundo cerrado
foil_use_negations(false).     % No usa información negativa
foil_det_lit_bound(0).         % No añade literales determinados
```

- **Ejemplos**

```
enlace(1,2).  enlace(2,3).  enlace(3,4).  enlace(3,5).
camino(1,2).  camino(1,3).  camino(1,4).  camino(1,5).
camino(2,3).  camino(2,4).  camino(2,5).
camino(3,4).  camino(3,5).
```

Ejemplo con FOIL de relación recursiva

- **Sesión:**

```
?- foil(camino/2).
Uncovered positives: [(1,2),(1,3),(1,4),(1,5),(2,3),(2,4),(2,5),(3,4),(3,5)]
Adding a clause ...
Specializing current clause: camino(A,B).
Covered negatives: [(1,1),(2,1),(2,2),(3,1),(3,2),(3,3),(4,1),(4,2),(4,3),
                    (4,4),(4,5),(5,1),(5,2),(5,3),(5,4),(5,5)]
Covered positives: [(1,2),(1,3),(1,4),(1,5),(2,3),(2,4),(2,5),(3,4),(3,5)]

Ganancia: -2.630 Cláusula: camino(A,B):-enlace(C,A)
Ganancia:  5.503 Cláusula: camino(A,B):-enlace(A,C)
Ganancia:  2.897 Cláusula: camino(A,B):-enlace(C,B)
Ganancia: -1.578 Cláusula: camino(A,B):-enlace(B,C)
Ganancia:  0.000 Cláusula: camino(A,B):-enlace(A,A)
Ganancia:  0.000 Cláusula: camino(A,B):-enlace(B,A)
Ganancia:  5.896 Cláusula: camino(A,B):-enlace(A,B)
Ganancia:  0.000 Cláusula: camino(A,B):-enlace(B,B)
```

Ejemplo con FOIL de relación recursiva

- **Sesión (continuación):**

```
Clause found: camino(A,B) :- enlace(A,B).
Uncovered positives: [(1,3),(1,4),(1,5),(2,4),(2,5)]
Adding a clause ...
Specializing current clause: camino(A,B).
Covered negatives: [(1,1),(2,1),(2,2),(3,1),(3,2),(3,3),(4,1),(4,2),(4,3),
                    (4,4),(4,5),(5,1),(5,2),(5,3),(5,4),(5,5)]
Covered positives: [(1,3),(1,4),(1,5),(2,4),(2,5)]

Ganancia: -2.034 Cláusula: camino(A,B):-enlace(C,A)
Ganancia:  2.925 Cláusula: camino(A,B):-enlace(A,C)
Ganancia:  1.962 Cláusula: camino(A,B):-enlace(C,B)
Ganancia: -1.017 Cláusula: camino(A,B):-enlace(B,C)
Ganancia:  0.000 Cláusula: camino(A,B):-enlace(A,A)
Ganancia:  0.000 Cláusula: camino(A,B):-enlace(B,A)
Ganancia:  0.000 Cláusula: camino(A,B):-enlace(A,B)
Ganancia:  0.000 Cláusula: camino(A,B):-enlace(B,B)
```

Ejemplo con FOIL de relación recursiva

- **Sesión (continuación):**

```
Specializing current clause: camino(A,B) :- enlace(A,C).
Covered negatives: [(1,1),(2,1),(2,2),(3,1),(3,2),(3,3)]
Covered positives: [(1,3),(1,4),(1,5),(2,4),(2,5)]

Ganancia:  7.427 Cláusula: camino(A,B):-enlace(A,C),camino(C,B)
Ganancia: -1.673 Cláusula: camino(A,B):-enlace(A,C),enlace(D,A)
Ganancia: -2.573 Cláusula: camino(A,B):-enlace(A,C),enlace(A,D)
Ganancia:  2.427 Cláusula: camino(A,B):-enlace(A,C),enlace(D,B)
Ganancia: -1.215 Cláusula: camino(A,B):-enlace(A,C),enlace(B,D)
Ganancia:  3.539 Cláusula: camino(A,B):-enlace(A,C),enlace(C,D)
Ganancia:  4.456 Cláusula: camino(A,B):-enlace(A,C),enlace(C,B)

Clause found: camino(A,B) :- enlace(A,C), camino(C,B).

Found definition:
camino(A,B) :- enlace(A,C), camino(C,B).
camino(A,B) :- enlace(A,B).
```

Bibliografía

- **Witten, I.H. y Frank, E. *Data mining* (Morgan Kaufmann Publishers, 2000)**
  - **Cap. 3: "Output: Knowledge representation"**
  - **Cap. 4: "Algorithms: The basic methods"**
- **Mitchell, T.M. *Machine Learning* (McGraw-Hill, 1997)**
  - **Cap. 10: "Learning Sets of Rules"**
- **Russell, S. y Norvig, P. *Inteligencia artificial (Un enfoque moderno)* (Prentice–Hall Hispanoamericana, 1996)**
  - **Cap. 21: "Aprendiendo conocimiento"**