

# Introducción a la Programación Lógica Inductiva

ILP = Inductive Logic Programming  
= machine learning  $\cap$  logic programming  
= aprender con lógica

Introduced by Muggleton in 1992

Traducida (y extendida) por Jacinto Dávila en 2012

# Aprendizaje (Computacional)

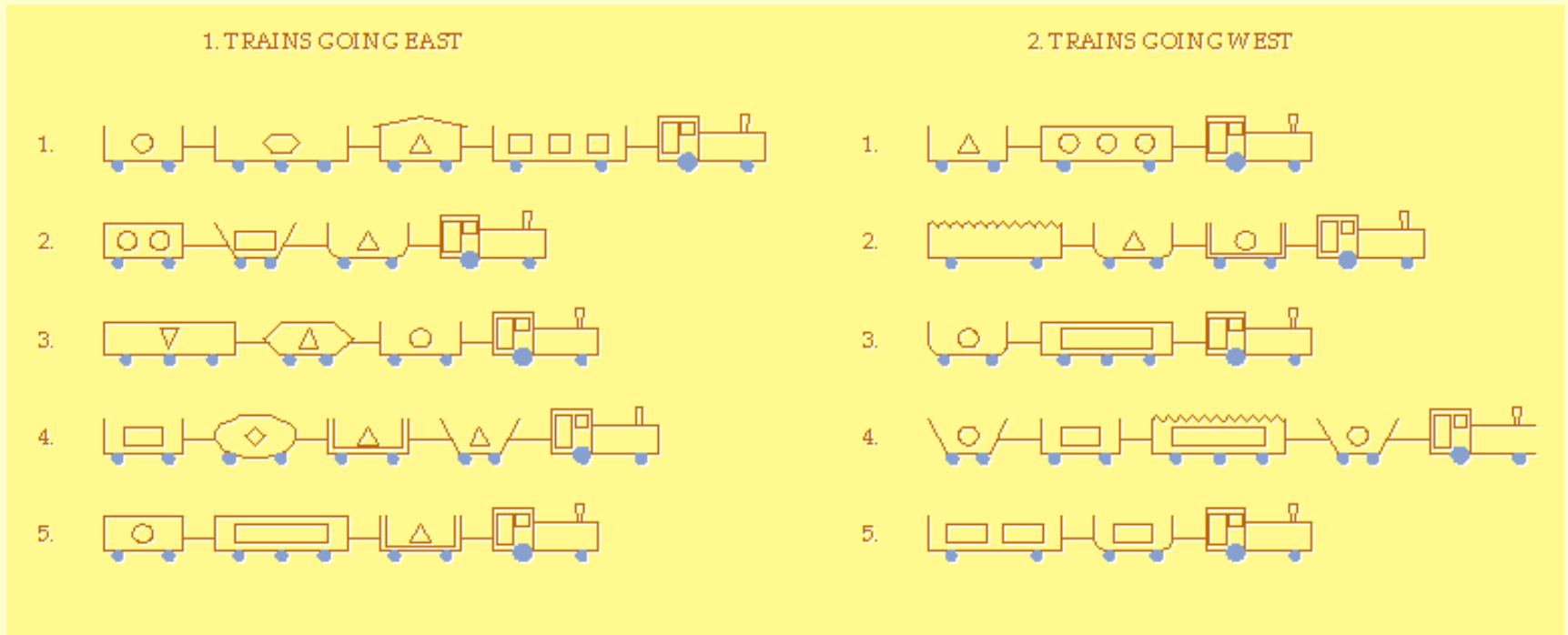
- El proceso por el cual ocurren cambios relativamente permanentes en el potencial conductual como resultado de la experiencia. (Anderson)
- Aprender es construir o modificar representaciones de lo que está siendo experimentado (Michalski)
- Se dice que un programa aprende de la experiencia E con respecto a alguna clase T de tareas y con una medida de rendimiento P, si su rendimiento en las tareas en T, medido con P, mejora con su experiencia E. (Mitchell)

# Técnicas de A. C.

- Aprendizaje de árboles de decisión
- Conceptual clustering
- Case-based learning
- Reinforcement learning
- Neural networks
- Genetic algorithms
- and... *Inductive Logic Programming*

# Por qué ILP ? - data con estructura

## Ejemplo seminal de los trenes este-oeste (Michalski)

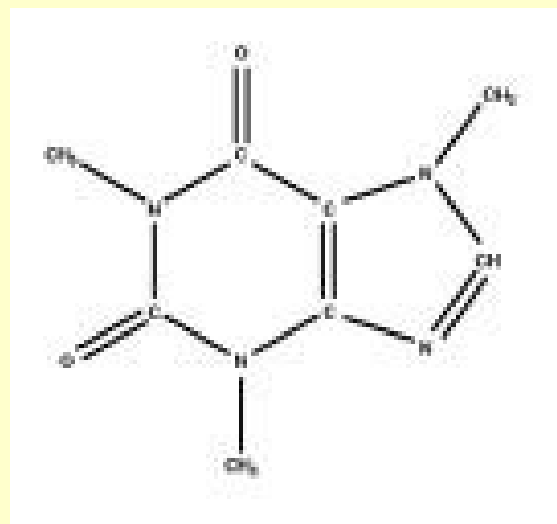
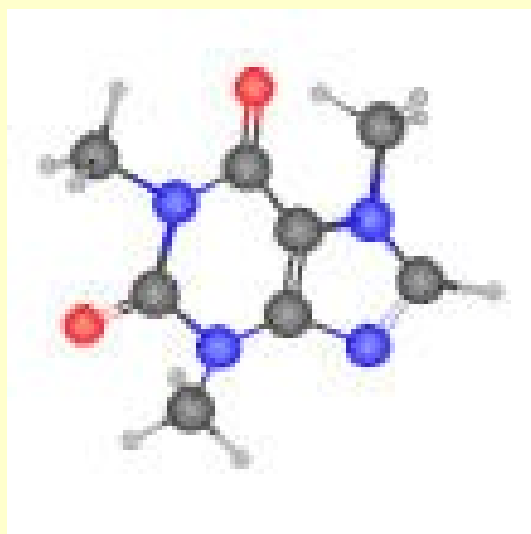


*¿Qué hace que un tren vaya hacia el este?*

# ¿Por qué ILP ? - data con estructura

Mutagenicity of chemical molecules

(King, Srinivasan, Muggleton, Sternberg, 1994)



¿Qué hace de una molécula mutagénica?

# ¿Por qué ILP ? – relaciones múltiples

## Propiedad relacionada con datos estructurados

has_car		car_properties					
Train	Car	Car	Length	Shape	Axes	Roof	...
t1	c11	c11	short	rectangle	2	none	...
t1	c12	c12	long	rectangle	3	none	...
t1	c13	c13	short	rectangle	2	peaked	...
t1	c14	c14	long	rectangle	2	none	...
t2	c21	c21	short	rectangle	2	flat	...
...	...	...	...	...	...	...	...

# ¿Por qué ILP ? – relaciones múltiples

Ejemplo de la genealogía:

- Dadas unas relaciones conocidas...
  - `papa(Viejo, Joven)` and `mama(viejo, Joven)`
  - `masculino(Alguno)` and `femenina(Alguna)`
- ...aprende nuevas relaciones (sexistas):
  - `padre(X, Y) :- papa(X, Y) .`
  - `padre(X, Y) :- madre(X, Y) .`
  - `hermano(X, Y) :-`  
`masculino(X), padre(Z, X), padre(Z, Y) .`

Muchas técnicas de AC no pueden con más de una relación a la vez, e.g. árboles de decisión, redes neuronales

# ¿Por qué ILP ? – bases lógicas

- Prolog = Programming with Logic  
puede representar:
  - *Background knowledge* (de un dominio): **hechos**
  - Ejemplos (de la relación a ser aprendida): **hechos**
  - Teorías (resultados del aprendizaje): **reglas**
- Soporta 3 formas de razonamiento lógico
  - Deducción
  - Inducción
  - Abducción



# Prolog - recordatorio

- Variables: `X`, `Y`, `Something`, `Somebody`
- Términos: `arthur`, `1`, `[1,2,3]`
- Predicados: `papá/2`, `female/1`
- Hechos:
  - `papa(christopher,victoria).`
  - `femenina(victoria).`
- Reglas:
  - `padre(X,Y) :- papa(X,Y).`

# Razonamiento lógico: deducción

De las reglas a los hechos...

B

∪

T

|-

E

mamá(penelope,victoria).  
mamá(penelope,arthur).  
papá(christopher,victoria).  
papá(christopher,arthur).

padre(penelope,victoria).  
padre(penelope,arthur).  
padre(christopher,victoria).  
padre(christopher,arthur).

padre(X,Y) :- papá(X,Y).  
padre(X,Y) :- mamá(X,Y).

# Razonamiento lógico: → inducción

De los hechos a las reglas... +/-

B

∪

E

|-

T

mother(penelope,victoria).  
mother(penelope,arthur).  
papá(christopher,victoria).  
papá(christopher,arthur).

padre(penelope,victoria).  
padre(penelope,arthur).  
padre(christopher,victoria).  
padre(christopher,arthur).

padre(X,Y) :- papá(X,Y).  
padre(X,Y) :- mother(X,Y).

# Inducción de un clasificador o Aprendizaje de Conceptos

*La tarea más estudiada en A.C.*

**Dado:**

- background knowledge **B**
- Un conjunto de ejemplos de entrenamiento **E**
- Una clasificación  $c \in \mathbf{C}$  para cada ejemplo  $e$

**Encuentra:** Una teoría **T** (o *hipótesis*) tal que

$$B \cup T \models c(e), \text{ para todo } e \in E$$

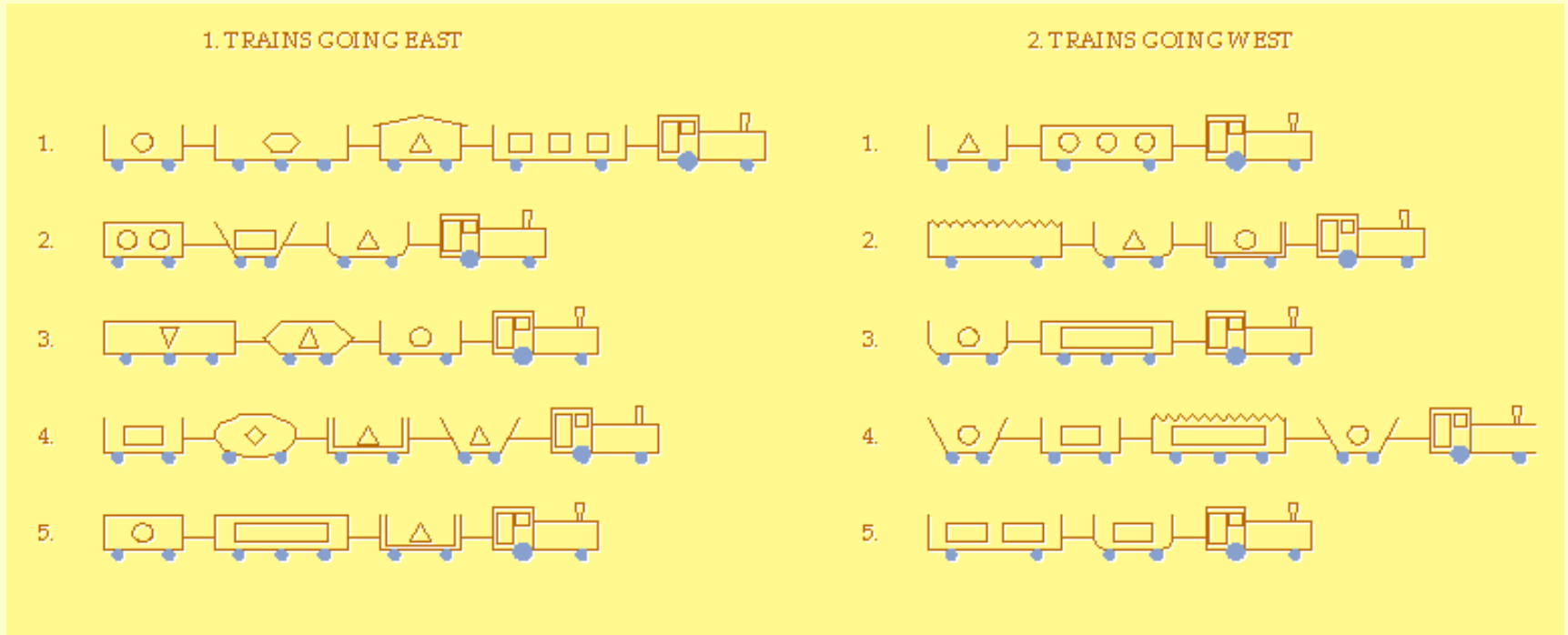
# Inducción de un clasificador: un ejemplo

El Ejemplo de los trenes hacia el Este

- **B:** las relaciones `has_car` y `car_properties` (`length`, `roof`, `shape`, etc.)  
e.g.: `has_car(t1, c11)`, `shape(c11, bucket)`
- **E:** Los trenes del `t1` al `t10` (con sus descripciones)
- **C:** `east`, `west`

# ¿Por qué ILP ? - data con estructura

## Ejemplo seminal de los trenes este-oeste (Michalski)



*¿Qué hace que un tren vaya hacia el este?*

# Inducción de un clasificador: un ejemplo

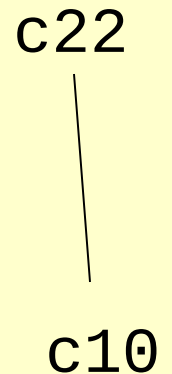
## El Ejemplo de los trenes hacia el Este

- **B:** las relaciones `has_car` y `car_properties` (`length`, `roof`, `shape`, etc.)  
e.g.: `has_car(t1, c11)`, `shape(c11, bucket)`
- **E:** Los trenes del `t1` al `t10` (con sus descripciones)
- **C:** `east`, `west`
- Posible **T**:  
`east(T) :-`  
    `has_car(T, C), length(C, short), roof(C, _).`

# Inducción de un clasificador: un ejemplo

## El ejemplo de la mutagénesis

- **B**: las relaciones atom y bond  
eg.: `atom(mol23, atom1, c, 195) .`  
`bond(mol23, atom1, atom3, 7) .`
- **E**: 230 moléculas con clasificación conocida
- **C**: active y nonactive con respecto a mutagénesis
- Posible **T**:  
`active(Mol) :-`  
    `atom(Mol, A, c, 22), atom(Mol, B, c, 10),`  
    `bond(Mol, A, B, 1) .`





# Aprender es buscar

## Dada:

- *Background knowledge*  $B$
- Un lenguaje para describir una teoría  $T$
- Ejemplos positivos  $P$  (clase +)
- Ejemplos negativos  $N$  (clase -)
- Una relación de cobertura  **$\text{covers}(B, T, e)$**

## Encuentra: Una teoría que cubra:

- Todos los ejemplos positivos (completitud)
- Ningún ejemplo negativo (consistencia)

# Aprender es buscar

- La relación de cobertura en ILP se define:  
$$\text{covers}(B, T, e) \Leftrightarrow B \cup T \vdash e$$
- Una teoría es un conjunto de reglas
- A cada regla se le busca por separado (eficiencia)
- Cada regla propuesta debe ser consistente (no cubrir negativos), pero no necesariamente completa
- Estrategia *divide y vencerás*
  - Elimine de P los ejemplos que ya cubren la reglas actuales en T

# La exploración del espacio

¿Alguna estrategia?

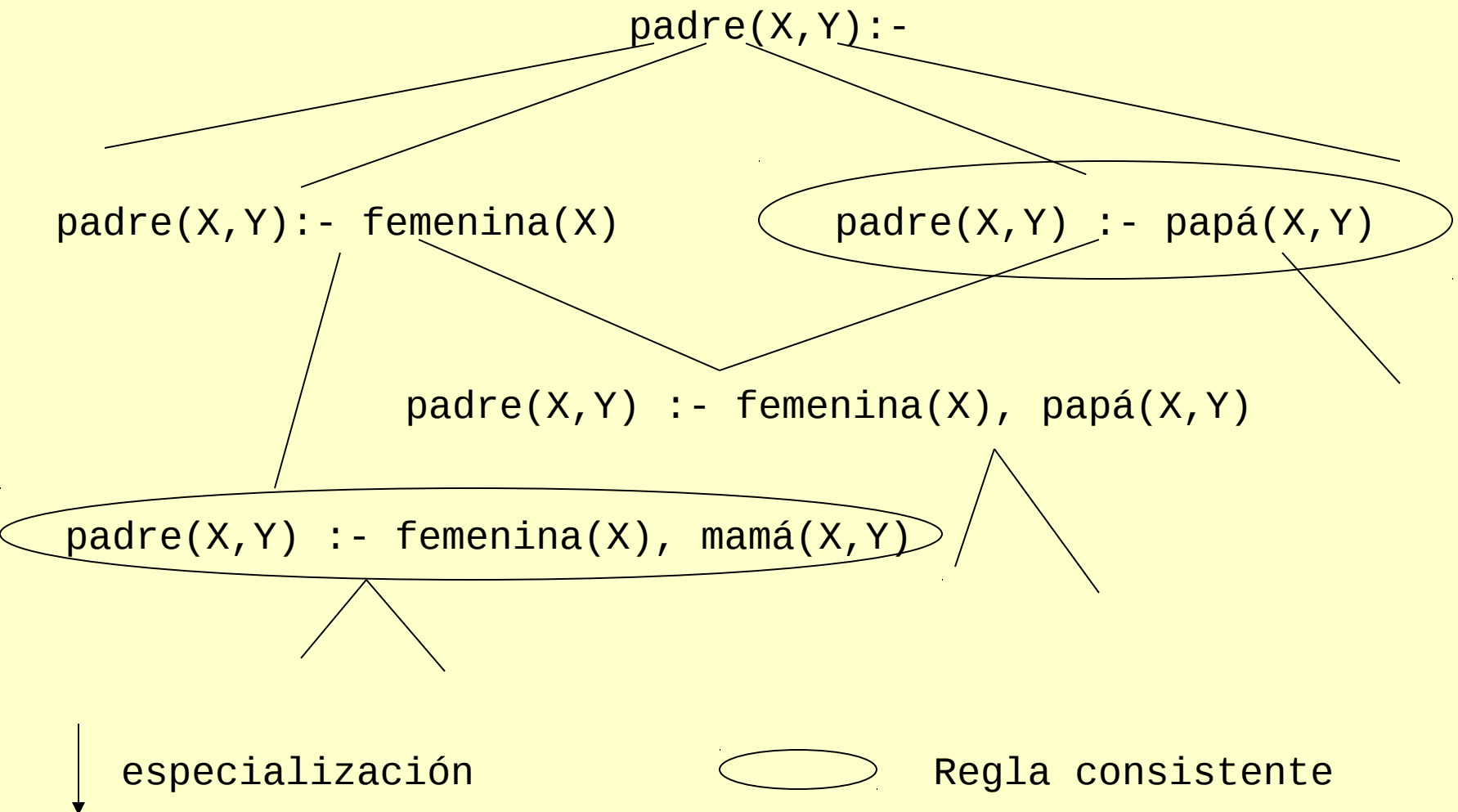
- *Random walk*
  - Redundancia, incompletitud en la búsqueda
- Hacerlo sistemático según algún orden:
  - Mejor control → no redundancia, completitud.
  - El ordenamiento puede ser usado para guiar la búsqueda hacia mejores reglas.

*Pero.. ¿Qué clase de ordenamiento?*

# Ordenamiento por generalidad

- La Regla 1 es más general que la regla 2  $\rightarrow$  la regla 1 cubre más ejemplos que la regla 2
  - Si una regla es consistente (no cubre negativos) entonces cualquier especialización suya también es consistente
  - Si una regla es completa (cubre todos los positivos) entonces toda generalización suya también lo es
- Un medio de podar el espacio de búsqueda
- Dos tipos de movimientos: Especialización y generalización
- Forma común de ordenamiento en:  *$\theta$ -subsumption*

# Ordenamiento por Generalidad



# Sesgos de búsqueda

*“El sesgo se refiere a cualquier criterio para escoger una generalización antes que otra, además de mantener la consistencia estricta con las instancias de entrenamiento observadas” (Mitchell)*

- Sirve para restringir el espacio de búsqueda (eficiencia)
- Guía la búsqueda (para un dominio particular)
- Hay distintos tipos de sesgos
  - El sesgo lingüístico
  - El sesgo de búsqueda
  - El sesgo estratégico

# El Sesgo Lingüístico

- Elección de predicados:

`roof(C, flat) ? roof(C) ? flat(C) ?`

- Tipos de predicados :

~~`east(T) :- roof(T), roof(C, 3)`~~

- Modos de los predicados :

~~`east(T) :- roof(C, flat)`~~

`east(T) :- has_car(T, C), roof(C, flat)`

- Discretización de valores numéricos

# El sesgo de búsqueda

La dirección de los movimientos en el espacio de búsqueda

- De arriba hacia abajo
  - Comienzo: la regla vacía ( $c(X) :- .$ )
  - movimientos: especializaciones
- De abajo hacia arriba
  - comienzo: La cláusula al fondo ( $\sim c(X) :- B.$ )
  - movimientos: generalizaciones
- Bi-direccional



# Sesgo Estratégico

Búsqueda heurística de la mejor regla

- Hill-climbing:
  - Preserva sólo una regla
  - Eficiente pero puede perder el “máximo global”
- Búsqueda “iluminada” (Beam search):
  - Preserva k reglas para saltar atrás
  - Menos agresiva
- Búsqueda primer el mejor:
  - Preserva todas las reglas
  - Es más costosa pero es completa!

# Un algoritmo ILP genérico

**procedure** ILP(*Ejemplos*)

  Iniciar(*Reglas*, *Ejemplos*)

**repeat**

*R* = Seleccionar(*Reglas*, *Ejemplos*)

*Rs* = Refina(*R*, *Ejemplos*)

*Reglas* = Reduce(*Reglas*+*Rs*, *Reglas*)

**until** CriterioDeParada(*Reglas*, *Ejemplos*)

**return**(*Reglas*)

# Un algoritmo ILP genérico

- $\text{Iniciar}(\text{Reglas}, \text{Ejemplos})$ : Inicializa un conjunto de teorías como puntos de arranque de la búsqueda
- $\text{Seleccionar}(\text{Reglas}, \text{Ejemplos})$ : Selecciona la regla candidata más prometedora  $R$
- $\text{Refina}(R, \text{Ejemplos})$ : Retorna la vecindad de  $R$  (usando especialización o generalización)
- $\text{Reduce}(\text{Reglas}, \text{Ejemplos})$ : Elimina teorías que no sirvan (todas salvo una en *hill-climbing*, ninguna en búsqueda primero el mejor)

Google Search: Inductive.Logic.Programming - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail News RSS Feeds

Address <http://www.google.com/search?hl=en&lr=&ie=ISO-8859-1&oe=ISO-8859-1&q=Inductive.Logic.Programming> Go Links »

Google Web Images Groups News Froogle <sup>New!</sup> more »

Inductive.Logic.Programming Search [Advanced Search](#) [Preferences](#)

**Web** Results 1 - 10 of about 24,500 for [Inductive.Logic.Programming](#). (0.40 seconds)

[ILPNet](#)  
Welcome to ILPNET. ILPNET is the **Inductive Logic Programming** European Scientific Network (network no. CP93-44), which was in 1993 ...  
[www-ai.ijs.si/ilpnet/](http://www-ai.ijs.si/ilpnet/) - 3k - [Cached](#) - [Similar pages](#)

[Inductive Logic Programming](#)  
**Inductive Logic Programming** Techniques and Applications. ... **Inductive Logic Programming:** Techniques and Applications. Ellis Horwood, New York, 1994. ...  
[www-ai.ijs.si/SasoDzeroski/ILPBook/](http://www-ai.ijs.si/SasoDzeroski/ILPBook/) - 5k - [Cached](#) - [Similar pages](#)  
[ [More results from www-ai.ijs.si](#) ]

[ILPnet2](#)  
... Network of Excellence in **Inductive Logic Programming** ILPnet2. Funded by the European Commission under contract INCO 977102. Administrative ...  
[www.cs.bris.ac.uk/~ILPnet2/](http://www.cs.bris.ac.uk/~ILPnet2/) - 10k - [Cached](#) - [Similar pages](#)

[Inductive Logic Programming](#)  
**Inductive Logic Programming.** ... Recommended Reading: Selected Chapters from Stephen Muggleton (ed.) **Inductive Logic Programming**, Academic Press, 1992. ...  
[www.ai.univie.ac.at/~juffi/Ilp-Vu/Ilp-vu-program.html](http://www.ai.univie.ac.at/~juffi/Ilp-Vu/Ilp-vu-program.html) - 9k - [Cached](#) - [Similar pages](#)

[ILP2000: Tenth International Conference on Inductive Logic ...](#)  
Tenth International Conference on. **Inductive Logic Programming** (ILP2000) July 24-27, 2000, London. ...  
[www.cs.york.ac.uk/ILP-events/ILP-2000/](http://www.cs.york.ac.uk/ILP-events/ILP-2000/) - 9k - [Cached](#) - [Similar pages](#)

[Esprit LTR 20237 Inductive Logic Programming II \(ILP2\) Home Page](#)  
Esprit LTR 20237 **Inductive Logic Programming II** (ILP2) Home Page. ... 20237 on **Inductive Logic Programming II** (ILP2). Coordinator, KULeuven. Partners. ...  
[www.cs.kuleuven.ac.be/~ml/esprit/esprit\\_ilp2\\_20237.html](http://www.cs.kuleuven.ac.be/~ml/esprit/esprit_ilp2_20237.html) - 6k - [Cached](#) - [Similar pages](#)

Start cs363 Microsoft PowerPoint - [I...] Google Search: Inducti... Internet 06:43

inductive logic programming - Buscar con Google - Mozilla Firefox

inductive logic programming - ...

Ir a un sitio web

Google

+Tú Búsqueda Imágenes Maps Play YouTube Noticias Gmail Docs Calendar Más ▾

Google

Inductive Logic Programming

Iniciar sesión

Búsqueda

Aproximadamente 665,000 resultados (0.16 segundos)

Todo

Imágenes

Maps

Videos

Noticias

Shopping

Más

Buscar en la Web

Buscar sólo páginas en español

Páginas extranjeras traducidas

Más herramientas

[Artículos académicos para inductive logic programming](#)

[Inductive logic programming](#) - Lavrac - Citado por 896

[Inductive logic programming](#) - Muggleton - Citado por 1228

[The application of Inductive Logic Programming to ...](#) - Dolsak - Citado por 154

[Inductive logic programming - Wikipedia, the free encyclopedia](#)

[en.wikipedia.org/.../Inductive\\_logic\\_programmi...](#) - Traducir esta página

**Inductive logic programming (ILP)** is a subfield of machine learning which uses logic programming as a uniform representation for examples, background ...

[Inductive Logic Programming](#)

[www.doc.ic.ac.uk/~shm/ilp.html](#) - Traducir esta página

**Inductive Logic Programming (ILP)** is a research area formed at the intersection of Machine Learning and Logic Programming. **ILP** systems develop predicate ...

[Inductive Logic Programming: Techniques and Applications - Institut](#)

[www-ai.ijs.si/SasoDzeroski/ILPBook/](#) - Traducir esta página

**Inductive Logic Programming:** Techniques and Applications. Ellis Horwood, New York, 1994. For a state-of-the art account of **inductive logic programming** ...

[Inductive Logic Programming - Theory](#)

[www.cs.ox.ac.uk/activities/.../ilp\\_theory.html](#) - Traducir esta página

**Inductive** inference based on inverting resolution in propositional **logic** was the basis of the **inductive** inference rules within the Duce system. Duce had six ...

**ILP 2012 122nd International Conference on Inductive Logic**

# ILPnet2 — [www.cs.bris.ac.uk/~ILPnet2/](http://www.cs.bris.ac.uk/~ILPnet2/)

## Network of Excellence in ILP in Europe

- 37 universities and research institutes
- Educational materials
- Publications
- Events (conferences, summer schools, ...)
- Description of ILP systems
- Applications

# Sistemas ILP

- FOIL (Quinlan and Cameron-Jones 1993): top-down hill-climbing search
- Progol (Muggleton, 1995): top-down best-first search with bottom clause
- Golem (Muggleton and Feng 1992): bottom-up hill-climbing search
- LINUS (Lavraac and Dzeroski 1994): propositionalisation
- Aleph (~Progol), Tilde (relational decision trees),
- Tertius (Flach, P. and Lachiche, N, 2001  
<http://www.cs.bris.ac.uk/Research/MachineLearning/Tertius/>)
- ...

# Aplicaciones ILP

<http://www.cs.ox.ac.uk/activities/machlearn/ilp.html>

- Ciencias de la vida
  - mutagénesis, predecir en toxicología
  - Plegado y estructura de proteína
  - Genómica
- Procesamiento del lenguaje natural
  - Tiempo pasado de verbos en inglés
  - Clasificación y análisis de documentos
- Ingeniería
  - Diseño de redes de elementos finitos
- Ciencias ambientales
  - biodegradabilidad de componentes químicos



# Fin

## Libros sobre ILP...

- Flach, Peter. Machine Learning: The Art and Science of Algorithms that Make Sense of Data. CUP. 2012  
En: <http://www.cs.bris.ac.uk/~flach/mlbook/>
- De Raedt, Luc. Logical And Relational Learning. Springer. 2008  
<http://www.springer.com/computer/ai/book/978-3-540-20040-6>
- J. Lloyd. **Logic for learning:**  
**learning comprehensible theories from structured data** 2003.
- Flach, Peter. Simply Logical (último capítulo):  
<http://www.cs.bris.ac.uk/~flach/SimplyLogical.html>
- Dávila, J. Lógica Práctica y Aprendizaje Computacional. EAE 2011.  
<http://webdelprofesor.ula.ve/ingenieria/jacinto/libros/logica-practica-aprendizaje-computacional.pdf>