Programación Declarativa: Lógica y Restricciones

nceptos Básicos de la Programación en Prolog

Mari Carmen Suárez de Figueroa Baonza

mcsuarez@fi.upm.es

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

nidos

- ficación
- uctruras de datos
- ursividad, *backtracking* y búsqueda
- trol de ejecución

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

...ación (I)

...nificación es el mecanismo que se encarga de ...ver las igualdades lógicas y de dar valor a las ...bles lógicas

- ...a unificación no se evalúan expresiones
- ...a evaluar expresiones existe un operador especial "*is*"
  - Antes de realizar la unificación evalúa la parte derecha como si se tratase de una expresión aritmética

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ación (II): Reglas Generales

y T2 son constantes, entonces T1 y T2 unifican si déntica s

. y T2 son variables, entonces T1 y T2 unifican pre

es una variable y T2 es cualquier tipo de término, nces T1 y T2 unifican y T1 se instancia con T2

y T2 son términos complejos, unifican si:

- nen el mismo functor y aridad
- os los argumentos unifican

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ación (III). Ejemplos

- *pe(A,rojo)=jose(B,rojo)?*
  - *epe!=jose → No unifica*
- *pe(A,rojo)=pepe(Z,rojo)?*
  - *epe=pepe, A=Z, rojo = rojo → Si unifican*
- *a?*
- *Y?*
- *)=f(X)?*
- *,a)=f(b,Y)?*
- *,a,X)=f(b,Y,c)?*
- *f(X)?*

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ación (IV): =/2

ıboliza el predicado *“unifica”*:

redicado =/2 está *predefinido* en ISO-Prolog, no es necesario gramar la unificación, que es una característica básica del nostrador automático.

dos argumentos son las dos expresiones a unificar.

e predicado es *verdadero* si sus dos argumentos unifican:

= a ; f(a,b) = f(a,b) ; X = a ; f(a,b) = X ; X = Y ; f(a,X) = f(a,b) ; X = f(Y)

s *falso* si sus dos argumentos no unifican:

= b ; f(a,b) = f(b,a) ; X = f(X) ; f(X,X) = f(g(Y),Y)

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ación implícita y la variable anónima

- ıificación de una variable puede realizarse:
  - lícitamente, empleando =/2 como un objetivo más
  - olícitamente, dado que Resolución = Corte + Unificación

```
n es amigo de cualquiera que sea rico"
rico(x) -> amigo(juan,x))
juan,X):- rico(X).
```

s necesario unificar el segundo argumento de amigo/2, para que su alor se emplee en la prueba de rico/1

- riable anónima:
  - ácticamente: '_'; '_Anónima'; '_X'
  - ıánticamente: no se unifica, no toma nunca valor

```
e es amigo de todo el mundo"
amigo(pepe,x)
pepe,_).
```

s innecesario unificar el segundo argumento de amigo/2, cualquier alor es aceptable y no se emplea en ulteriores objetivos

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

de la Unificación en Ejecución

of Unification in Execution

s mentioned before, unification used to *access data* and *give values to variables*.

ample: Consider query `?- animal(A), named(A,Name).` with:

```
imal(dog(barry)).
med(dog(Name),Name).
```

xecution of `animal(A)` assigns a (ground) value to `A`.

xecution of `named(A,Name)` assigns a (ground) value to `Name` by accessing the ta in the subfield of the `dog/1` structure.

so, unification is used to *pass parameters* in procedure calls and to *return lues* upon procedure exit.

`- animal(A), named(A,Name)`	returns a value upon exit of `animal(A)`
`- named(dog(barry),Name)`	passes a value in first argument of call to `named/2`
`ame = barry`	returns a value in second argument on exit of `named/2`

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

s de uso

Nota: En la definición de un procedimiento no hay parámetros predefinidos de "entrada" y/o "salida". El modo de uso de cada parámetro depende de la llamada o pregunta que se haga en cada momento al procedimiento.

ct, argument positions are not fixed a priory to be input or output.

mple: Consider query `?- pet(spot).` vs. `?- pet(X).`

h a call to a procedure, any argument may be ground, free, or partially intiated.

s, procedures can be used in different **modes** erent sets of arguments are input or output in each mode).

mple: Consider the following queries:

`amed(dog(barry),Name).` **% entrada, salida** `?- named(A,barry).` **% salida, entrada**

`amed(dog(barry),barry).` **% entrada, entrada** `?- named(A,Name).` **% salida, salida**

rgument may even be both input and output.

mple: Consider query `?- struct(f(A,b)).` with:

`ct(f(a,B)).`

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

o a los datos (I)

sing Data

esing subfields of *records*:

mple:

```
(date(Day,_Month,_Year),Day).
th(date(_Day,Month,_Year),Month).
r(date(_Day,_Month,Year),Year).
```

ming subfields:

mple:

```
e(day,  date(Day,_Month,_Year),Day).
e(month,date(_Day,Month,_Year),Month).
e(year, date(_Day,_Month,Year),Year).
```


CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

o a los datos (II)

sing Data (Contd.)

alizing variables:

mple: `?- init(X), ...`

```
(date(9,6,2011)).
```

paring values:

mple: `?- init_1(X), init_2(Y), equal(X,Y).`

```
l(X,X).
```

mply: `?- init_1(X), init_2(X).`

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Estructurados (I)

ured Data and Data Abstraction (and the '=' Predicate)

a *structures* are created using (complex) terms.

ucturing data is important:

```
rse(complog,wed,18,30,20,30,'F.','Bueno',new,5102).
```

course/10

en is the Computational Logic course?

```
course(complog,Day,StartH,StartM,FinishH,FinishM,C,D,E,F).
```

uctured version:

```
course(complog,Time,Lecturer, Location) :-
    Time = t(wed,18:30,20:30),
    Lecturer = lect('F.','Bueno'),
    Location = loc(new,5102).
```

course/4

e: "X=Y" is equivalent to "=(X,Y)"

ere the predicate =/2 is defined as the fact "=(X,X)." – Plain unification!

uivalent to:

```
rse(complog,  t(wed,18:30,20:30),
     lect('F.','Bueno'),  loc(new,5102)).
```

Estructurados (II)

ured Data and Data Abstraction (and The Anonymous Variable)

en:

```
course(complog,Time,Lecturer, Location) :-
        Time = t(wed,18:30,20:30),
        Lecturer = lect('F.','Bueno'),
        Location = loc(new,5102).
```

en is the Computational Logic course?

```
course(complog,Time, A, B).
```

solution:

```
me=t(wed,18:30,20:30), A=lect('F.','Bueno'), B=loc(new,5102)}
```

ng the *anonymous variable* ("_"):

```
course(complog,Time, _, _).
```

solution:

```
me=t(wed,18:30,20:30)}
```


CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

cturas de Datos

sics_prolog.pdf

nsparencias 10-19

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

: Ejercicios

dromos

inir el predicado palindromo/1 tal que:

alindromo(X) es cierto si la lista X es palíndromo, es decir, puede leerse de misma manera al derecho y al revés

jemplos: palindromo([r,o,t,o,r]) es verdadero

palindromo([r,o,t,a,r]) es falso

palindromo([r,o,t|X]) es verdadero con {X = [o,r]}, o {X = [t,o,r]} o {X = [_A,t,o,r]} o ...

ero y Último

inir el predicado primeroultimo/1 tal que:

rimeroultimo(X) es cierto si el primer y ultimo elementos de la lista X son el ismo

jemplos: primeroultimo([a]) es verdadero

primeroultimo([a,f,t]) es falso

primeroultimo([X,f,t,a]) es verdadero con {X = a}

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

racking (I)

rama lógico:

e de conocimientos donde se expresan los hechos y las reglas de ucción de un dominio o problema

tor de inferencia que aplica el algoritmo de resolución. Este oritmo permite inferir nuevos datos relativos al mundo que amos representando

oma como entrada la base de conocimientos y el objetivo planteado y frece como salida un resultado de verdadero o falso en función de si ha odido o no demostrar el objetivo según la base de conocimientos

ste algoritmo se basa en el uso de la técnica de ***backtracking***, de forma que inferencia del objetivo plantedo se realiza a base de prueba y error

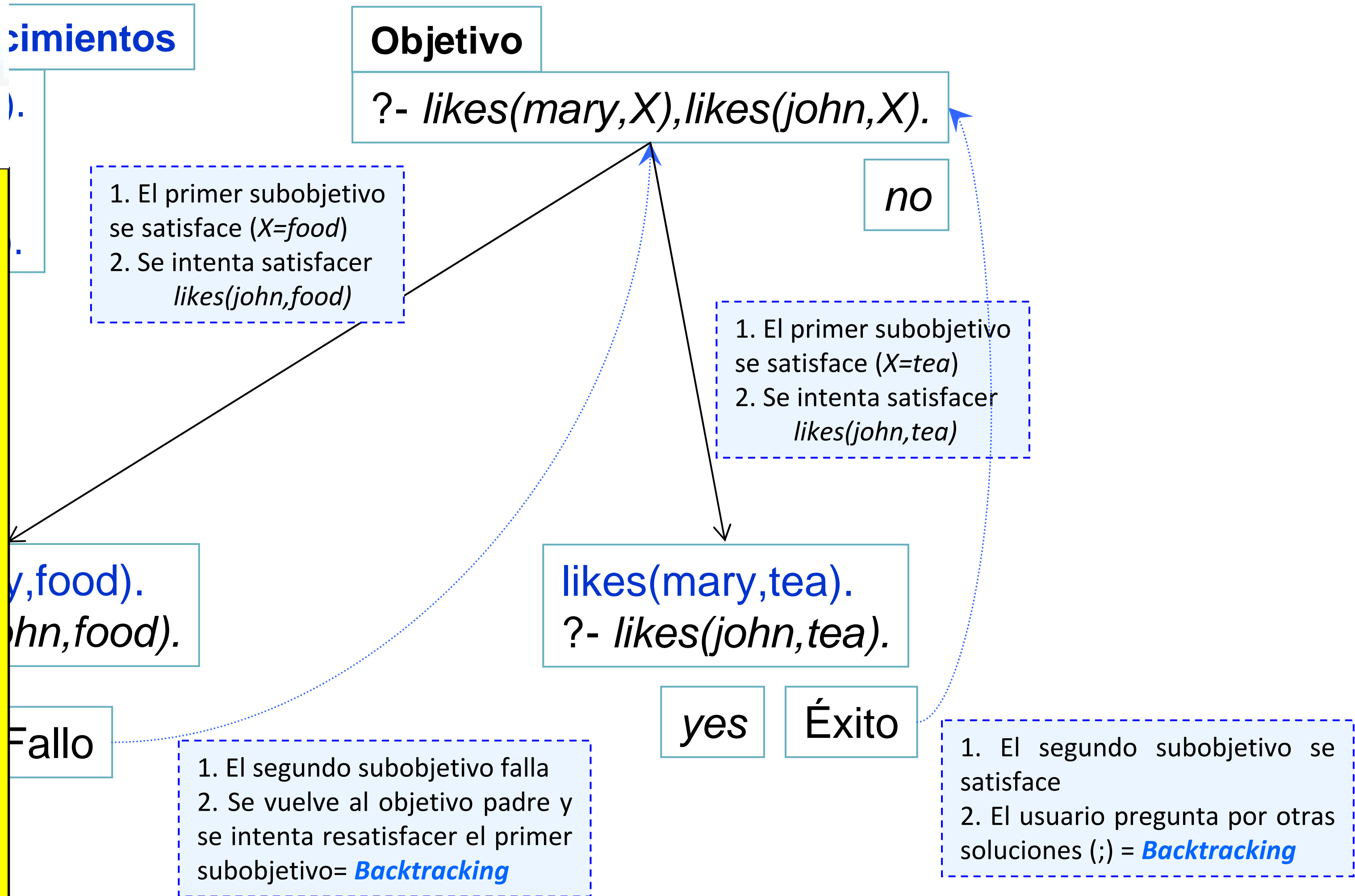
Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

racking (II)

- hecho puede hacer que un objetivo se cumpla diatamente
- regla sólo puede reducir la tarea a la de satisfacer conjunción de subobjetivos
- se puede satisfacer un objetivo, se iniciará un eso de *backtracking*
  - e proceso consiste en intentar satisfacer los objetivos buscando forma alternativa de hacerlo
- ecanismo de *backtracking* permite explorar los entes caminos de ejecución hasta que se encuentre olución
  - *ktracking* por fallo
  - *ktracking* por acción del usuario

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

racking (III). Ejemplo



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ol de la búsqueda

en 3 formas principales de controlar la ejecución de
ograma lógico

- rden de los literales en el cuerpo de una cláusula
- rden de las cláusulas en un predicado
- operadores de poda (ej., '*cut*')

den de los literales en una cláusula o el orden de las
ulas en el programa son importantes

- orden afecta tanto al correcto funcionamiento del programa, no al recorrido del árbol de llamadas, determinando, entre otras as, el orden en que Prolog devuelve las soluciones a una gunta dada

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

n de las cláusulas

den de las cláusulas determina el orden en que se nen las soluciones ya que varía la manera en que se rren las ramas del árbol de búsqueda de soluciones

árbol de búsqueda tiene alguna rama infinita, el n de las sentencias puede alterar la obtención de las iones, e incluso llegar a la no obtención de ninguna ión

n heurística: es recomendable que los hechos ezcan antes que las reglas del mismo predicado

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

n de las cláusulas: Ejemplo (I)

versiones de miembro de una lista (miembro(X,L)).
bas versiones tienen las mismas cláusulas pero
itas en distinto orden

ión 1:

```
miembro(X,[X|_]).
miembro(X,[_|Z]):- miembro(X,Z).
```

ión 2:

```
miembro(X,[_|Z]):- miembro(X,Z).
miembro(X,[X|_]).
```

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

n de las cláusulas: Ejemplo (II)

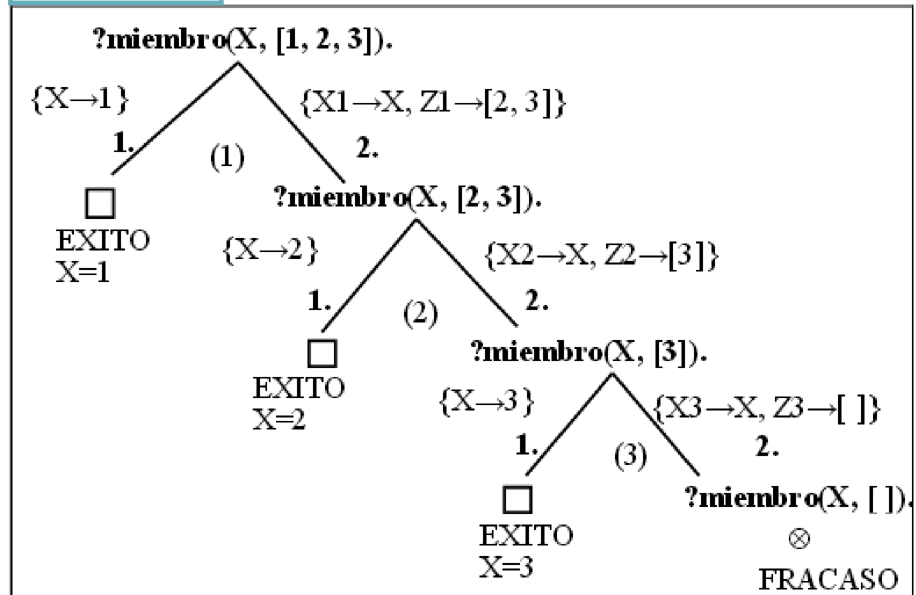
bas versiones les hacemos la misma pregunta

iembro (X, [1,2,3]).

ón 1:

nbro(X,[X|_]).

nbro(X,[_|Z]):- miembro(X,Z).



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

...n de las cláusulas: Ejemplo (III)

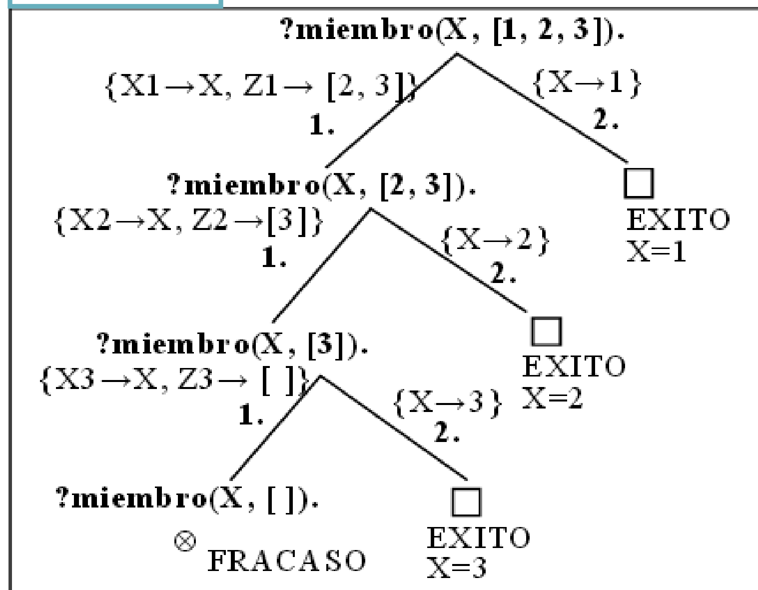
...bas versiones les hacemos la misma pregunta

...niembro (X, [1,2,3]).

...ón 2:

...nbro(X,[_|Z):- miembro(X,Z).

...nbro(X,[X|_]).



Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

n de los literales (I)

den de los literales dentro de una cláusula afecta al
cio de búsqueda y a la complejidad de los cómputos
os

ntas opciones en el orden de los literales pueden ser
ribles para distintos modos de uso

(X, Y) :- hombre(X), padre(Y, X).

ara modo (in, out): Se comprueba primero que el X es hombre y después
e busca a su padre Y

(X, Y) :- padre(Y, X), hombre(X).

ara modo (out, in): Se buscan los hijos de Y y después se seleccionan si son
ombres

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

n de los literales (II)

den de los literales en el cuerpo de una regla influye ién en la terminación

ersa([ ], [ ]).

ersa([C|R], Z) :- inversa(R, Y), concatenar(Y, [C], Z).

ara preguntas en modo (in, out) termina

ara preguntas en modo (out, in) el árbol de búsqueda tiene una rama nfinita, por lo que tras dar la respuesta correcta se queda en un bucle

ué sucede si intercambiamos los literales en la regla?

ales.pl

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70
Cartagena99

ol con poda: *cut* (I)

g proporciona un predicado predefinido llamado !/0) que influye en el comportamiento procedural s programas

rincipal función es reducir el espacio de búsqueda ndo dinámicamente el árbol de búsqueda

rte puede usarse:

a aumentar la eficiencia

e eliminan puntos de *backtracking* que se sabe que no pueden producir inguna solución

a modificar el comportamiento del programa

e eliminan puntos de *backtracking* que pueden producir soluciones válidas.
e implementa de este modo una forma débil de negación

ste tipo de corte debe utilizarse lo menos posible

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ol con poda: *cut* (II)

- edicado cut como objetivo se satisface siempre y no e re-satisfacerse
- so permite podar ramas del árbol de búsqueda de iones
- consecuencia, un programa que use el corte será ralmente más rápido y ocupará menos espacio en oria (no tiene que recordar los puntos de *tracking* para una posible reevaluación)
  - perador de corte '!' limita el *backtracking*
  - ndo se ejecuta el operador de corte elimina todos los puntos de *ktracking* anteriores dentro del predicado donde está definido luido el propio predicado)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

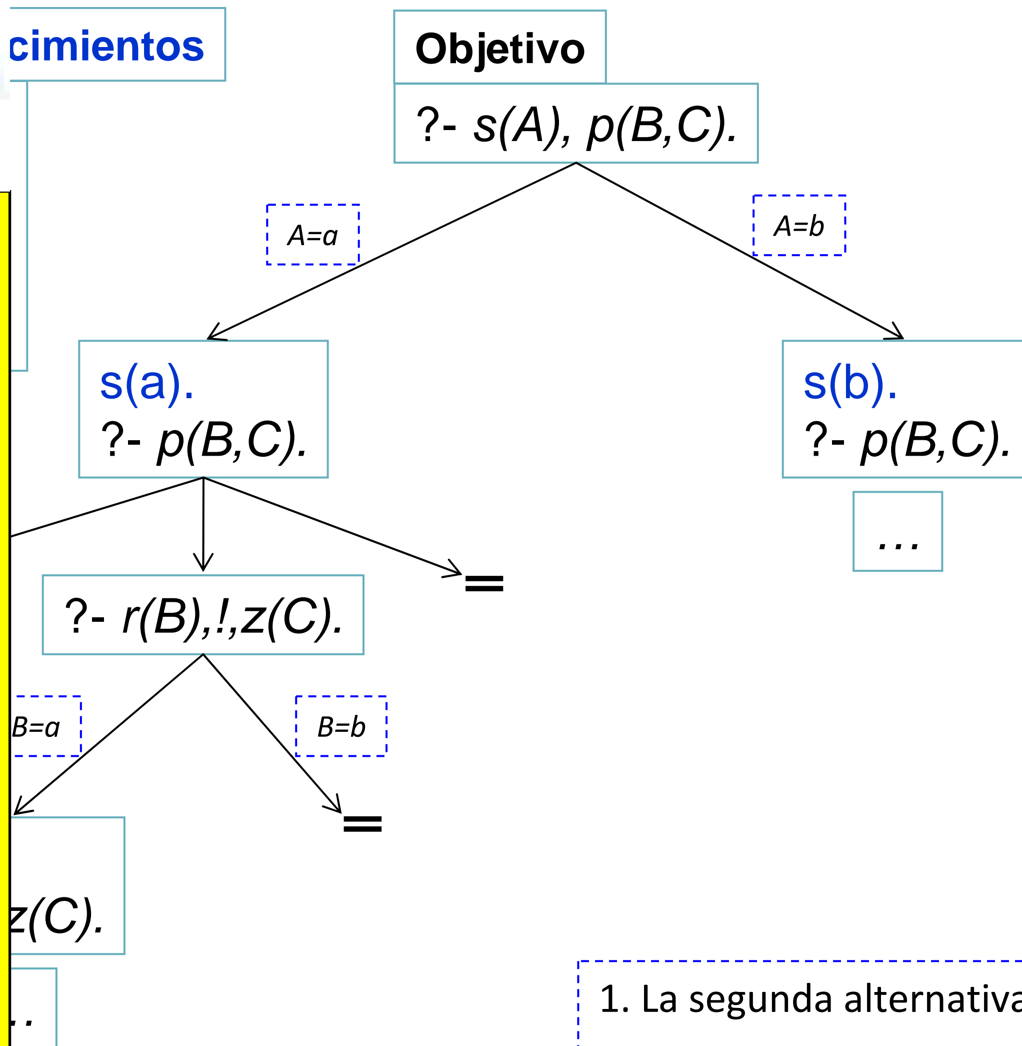
ol con poda: *cut* (III)

ncionamiento implica que:

- corte poda todas las alternativas correspondientes a cláusulas debajo de él
- corte poda todas las soluciones alternativas de la conjunción de etivos que aparezcan a su izquierda en la cláusula
  - sto es, una conjunción de objetivos seguida por un corte producirá como náximo una solución
- corte no afecta a los objetivos que estén a su derecha en la usula
  - stos objetivos pueden producir más de una solución, en caso de *acktracking*
  - in embargo, una vez que esta conjunción fracasa, la búsqueda continuará a artir de la última alternativa que había por encima de la elección de la entencia que contiene el corte

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ol con poda: *cut* (IV). Ejemplo



1. La segunda alternativa de r/1 (*r(b)*) no se considera
2. La tercera cláusula de p/2 no se considera

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ol con poda: *cut* (V)

miendo, de forma general, el efecto de un corte en regla C de la forma A :- B1, ..., Bk, !, Bk+2, ..., Bn, es uiente:

l objetivo actual G se unifica con A y los objetivos B1, ..., Bk se sfacen, entonces el programa fija la elección de esta regla para ucir G; cualquier otra regla alternativa para A que pueda ficarse con G se ignora

emás, si los Bi con i>k fracasan, la vuelta atrás sólo puede erse hasta el corte. Las demás elecciones que quedaran para nputar los Bi con i≤ k se han cortado del árbol de búsqueda

l *backtracking* llega de hecho al corte entonces éste fracasa y la queda continúa desde la última elección hecha antes de que G iera la regla C

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ol con poda: *cut*. Ejemplo (I)

cla (L1, L2, L), en modo (in,in,out), mezcla dos listas
enadas de números L1 y L2 en la lista ordenada L

```
mezcla ([X| Xs], [Y| Ys], [X| Zs]) :- X<Y, mezcla (Xs, [Y| Ys], Zs).
mezcla ([X| Xs], [Y| Ys], [X,Y | Zs]) :- X=Y, mezcla (Xs, Ys, Zs).
mezcla ([X| Xs], [Y| Ys], [Y| Zs]) :- X>Y, mezcla ([X| Xs], Ys, Zs).
mezcla (Xs, [ ], Xs).
mezcla ([ ], Ys, Ys).
```

nezcla de dos listas ordenadas es una operación
erminista

o una de las cinco cláusulas se aplica para cada objetivo (no
vial) en una computación dada

concreto, cuando comparamos dos números X e Y, sólo una de
tres comprobaciones X<Y, X=Y ó X>Y es cierta

a vez una comprobación se satisface, no existe posibilidad de
e alguna otra comprobacion se satisfaga

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ol con poda: *cut*. Ejemplo (II)

corte puede usarse para expresar la naturaleza uamente exclusiva de las comprobaciones de las primeras cláusulas

mezcla ([X | Xs], [Y | Ys], [X | Zs]) :- X<Y, !,mezcla (Xs, [Y | Ys], Zs).

mezcla ([X| Xs], [Y| Ys], [X, Y| Zs]) :- X=Y, !,mezcla (Xs, Ys, Zs).

mezcla ([X| Xs], [Y| Ys], [Y| Zs]) :- X>Y, !, mezcla ([X| Xs], Ys, Zs).

otra parte, los dos casos básicos del programa usulas 4 y 5) son también deterministas

mezcla (Xs, [ ], Xs):- !.

mezcla ([ ], Ys, Ys).

cláusula correcta a utilizar se elige por unificación con la cabeza, r eso el corte aparece como el primer objetivo (en este caso el ico) en el cuerpo de la cláusula 4. Dicho corte elimina la solución dundante (se volvería a obtener con la cláusula 5) al objetivo "?- ezcla([ ], [ ], X)"

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

de Corte

es: descartan soluciones correctas que no son sarias

afectan al sentido declarativo del programa

afectan a la eficiencia del programa

ctan a la completitud pero no a la corrección

an ramas inútiles, redundantes o infinitas

: descartan soluciones que no son correctas

ctan a la semántica declarativa del programa

difican el significado lógico del programa

liminar el operador de corte se obtiene un programa incorrecto

an soluciones erróneas podando ramas que conducen a éxitos deseados

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

s Verdes

teran el significado declarativo del programa

n programa semánticamente correcto se añade el
para obtener un programa más eficiente

ralmente se usan para expresar determinismo

arte del cuerpo que precede al corte (o a veces el patrón de la
eza) comprueba un caso que excluye a todos los demás

s:

```
ress(X,Add):- home_address(X,Add), !.
ress(X,Add):- business_address(X,Add).

mbercheck(X,[X|Xs]):- !.
mbercheck(X,[Y|Xs]):- membercheck(X,Xs).
```

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

s Verdes. Ejemplo (I)

te verde para evitar soluciones redundantes

```
_padre(X):-padre(X,Y),!.
dre(antonio,juan).
dre(antonio,maria).
dre(antonio,jose).
```

Corte.pl

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

s Verdes. Ejemplo (II)

e verde para evitar búsquedas inútiles

predicado ordenar(L,R) indica que R es el resultado de ordenar lista L por medio de intercambios sucesivos. Este predicado ará ordenada(L) que nos dice si la lista L está ordenada.

```
ordenar (L, R) :- concatenar(P, [X,Y | S], L), X>Y, !, concatenar(P, [Y,X| S],
NL), ordenar(NL, R).

ordenar (L, L) :- ordenada(L).
```

sabe que sólo hay una lista ordenada. Por tanto, no tiene ntido buscar otras alternativas una vez se ha encontrado la lista denada

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

s Rojos

an a la semántica declarativa del programa

n emplearse con cuidado

plos:

```
x(X,Y,X) :- X>Y, !.
x(X,Y,Y).
- max(5,2,2).

genitores(adan,0):-!.
genitores(eva,0):-!.
genitores(P,2).
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

cios (I): Uso del Corte

ar todas las apariciones de un cierto elemento en ista dada

rar(X, L1, L2)

2 es la lista obtenida al borrar todas las apariciones de X en la lista L1

Modo de uso (in, in, out), es decir, primer y segundo parámetros de entrada tercer parámetro de salida

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

cios (II): Uso del Corte

ir un predicado agregacion/3

egacion (X,L,L1): L1 es la lista obtenida añadiendo el elemento X lista L (si X no pertenece a L), y es L en caso contrario

- agregacion(a,[b,c[,L).

=[a,b,c])

-agregacion(b,[b,c],L).

=[b,c]

Cartagena99
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

cios (III): Uso del Corte

probar si una lista es sublista de otra

lista(X, Y)

es sublista de Y

lodo de uso (in, in)

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

amas *Generate & Test*

básicamente programas que generan soluciones idatas que se evalúan para comprobar si son o no ctas

algunas ocasiones es más sencillo comprobar si algo es una ución a un problema que crear la solución a dicho problema

isten en dividir la resolución de problemas en dos s:

erar soluciones candidatas

tear que las soluciones sean correctas

rogramas con la siguiente estructura:

serie de objetivos generan posibles soluciones vía *backtracking*

os objetivos comprueban si dichas soluciones son las apropiadas

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

amas *Generate & Test*: Ejemplo

nación de listas

enacion(X,Y)

es la lista resultante de ordenar la lista X de forma ascendente

a lista Y contiene, en orden ascendente, los mismos elementos que la lista

a lista Y es una permutación de la lista X con los elementos en orden scendente

ordenacion([2,1,2,3], L).

= [1,2,2,3]

enacion(X,Y) :-

ermutacion(X,Y), **(1) Generador: se obtiene una permutación de X en Y que pasa al objetivo (2) para comprobar si Y está ordenada**

rdenada_ascendente(Y). **(2) Prueba: comprueba si la lista está ordenada. Si no lo está, el *backtracking* se encarga de re-satisfacer el objetivo (1) buscando una nueva permutación**

enerateTest.pl

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

amas *Generate & Test*: Ejercicio

ir el predicado numeroParMenor/2 que es adero cuando X es un numero par menor que N

umeroParMenor(X,5).

=0; X=2; X=4

umeroParMenor(2,4).

es

umeroParMenor(3,5).

o

umeroParMenor(10,7).

o

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Programación Declarativa: Lógica y Restricciones

nceptos Básicos de la Programación en Prolog

Mari Carmen Suárez de Figueroa Baonza
mcsuarez@fi.upm.es