

Prolog

Control de la parte procesal de un programa

Laboratorio de Introducción a los Sistemas Informáticos Inteligentes
Escuela Superior de Ingeniería Informática - Universidad Vigo

Nota:

Basado en el capítulo 5 del libro [Prolog: Programming for Artificial Intelligence](#) de Ivan Bratko - Ed. Pearson - contenido en la bibliografía recomendada de la asignatura.

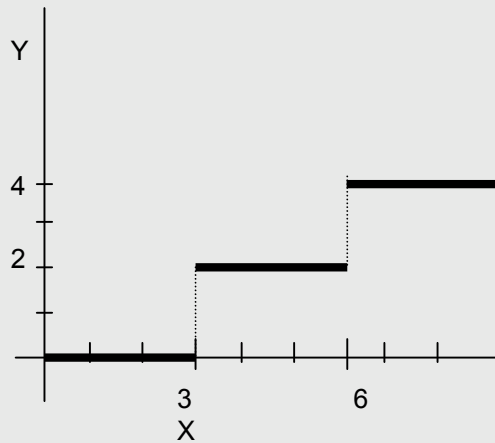
Contenido

- Proceso de vuelta atrás (backtracking)
- Ejemplos del uso del corte (cut)
- Negación por fallo

Proceso de vuelta atrás (backtracking)

- Prolog vuelve automáticamente atrás si es necesario para satisfacer un objetivo.
 - Elimina la necesidad de que el programador controle explícitamente este proceso
 - En algunos casos puede dar lugar a programas ineficientes

- Se puede controlar el backtracking haciendo uso de la operación de corte (cut)



- Ejemplo sencillo

```
f(X,0) :- X < 3.           % Regla1
f(X,2) :- 3=<X, X<6.       % Regla2
f(X,4) :- 6=<X.           % Regla3
```

- Nota: Se supone que antes de la ejecución X está instanciado a algún valor

Ejemplos del uso del corte (cut)

- Analicemos el ejemplo anterior

- Si se realiza la pregunta

```
?- f(1,Y), 2<Y.
```

- Usando la regla 1 del ejemplo el objetivo falla.
- Se intentan las otras reglas cuando el objetivo no se puede satisfacer

- Se podría mejorar el ejemplo usando el **corte**

- Su sintaxis es **!**
- Se inserta entre los subobjetivos
- Cuando se alcanza el corte no se realiza backtracking

```
f(X,0) :- X < 3, !.           % Regla1  
f(X,2) :- 3=<X, X<6, !.      % Regla2  
f(X,4) :- 6=<X.              % Regla3
```

Ejemplos del uso del corte (cut)

- En el programa modificado, hay otros tipos de ineficiencia
 - Si se realiza la pregunta

```
?- f(7,Y).
```

```
Y=4
```

- Para las reglas 1 y 2, no se alcanza el corte. Se hará backtracking hasta alcanzar la regla 3
- Cuando falla el subobjetivo de la regla 1, sabemos que tendrá éxito el de la regla siguiente, puesto que es la negación del anterior

- Se podría mejorar el ejemplo omitiendo los subobjetivos redundantes

```
si X<3 entonces Y=0,  
    en caso contrario si X < 6  
        entonces Y = 2,  
        en caso contrario Y=4
```

```
f(X,0) :- X < 3, !.           % Regla1  
f(X,2) :- X<6, !.           % Regla2  
f(X,4) .                     % Regla3
```

Ejemplos del uso del corte (cut)

- Otros ejemplos del uso de cut

- Definición del máximo

```
max(X,Y,X) :- X >=Y, !.
```

```
max(X,Y,Y).
```

- Nueva definición de miembro de una lista.

```
miembro(X, [X|L]) :- !.
```

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

Ejecución

```
?- miembro(X, [a,b,c]).  
X = a;  
no
```

Negación por fallo

- En Prolog existe un predicado especial **fail**, que es siempre falso, forzando así al padre a fallar.
- Ejemplo.

```
le_gusta(maria, X):- serpiente(X), !, fail.  
le_gusta(maria, X):- animal(X).
```

- Otro predicado importante en Prolog es **not**.
 - not(objetivo) es verdadero si objetivo no lo es.
 - La definición de not en Prolog sería:

```
not(P) :-  
    P, !, fail  
    ;  
    true
```