

Lenguajes de programación

Otras herramientas en PROLOG

Uso de la cláusula cut (!)

- Sirve para modificar el proceso de backtracking, cortando ramas del árbol de búsqueda de soluciones que ya no se necesitan recorrer.
- Ayuda en la eficiencia de ejecución de un programa en Prolog.
- Útil cuando basta una sola solución, o cuando hay riesgo de caer en una búsqueda infinita de soluciones.



Formato del cut (!)

- Sólo se utiliza el símbolo **!** en el cuerpo de una regla.
- Es una cláusula que es verdadera por default.
- Al ejecutarse, provoca que en el backtraking ya no se busquen alternativas de éxito para los términos anteriores al cut, y para la cabeza de la propia regla.



Ejemplo general

a :- $b, c, d, !, e, f.$

- El cumplimiento de a , depende del cumplimiento de b, c, d, e, f , pero en el proceso de búsqueda de soluciones alternas, ya no se ejecutan para d, c, b y a .

¿Cómo trabaja el Cut?

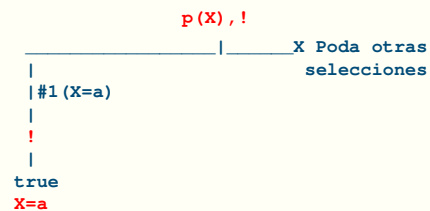
- Para ilustrar como produce Prolog las respuestas para programas y metas, considera el siguiente programa.

```
/* programa P */
p(a). /* #1 */
p(X) :- q(X), r(X). /* #2 */
p(X) :- u(X). /* #3 */
q(X) :- s(X). /* #4 */
r(a). /* #5 */
r(b). /* #6 */
s(a). /* #7 */
s(b). /* #8 */
s(c). /* #9 */
u(d). /* #10 */
```

- Si la consulta se estableciera como:
?- p(X), !.

X=a ;
No

- Generaría el siguiente árbol de derivación:



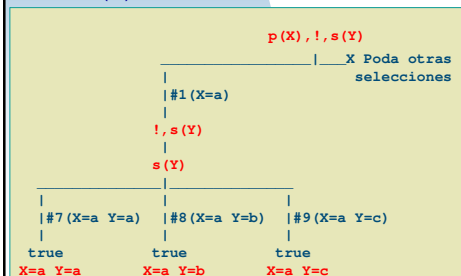
```
/* programa P */
p(a). /* #1 */
p(X) :- q(X), r(X). /* #2 */
p(X) :- u(X). /* #3 */
q(X) :- s(X). /* #4 */
r(a). /* #5 */
r(b). /* #6 */
s(a). /* #7 */
s(b). /* #8 */
s(c). /* #9 */
u(d). /* #10 */
```

Otros ejemplos de cut (!)

- Otras consultas:
?- p(X), !, s(Y).

X=a Y=a ;
X=a Y=b ;
X=a Y=c ;
No

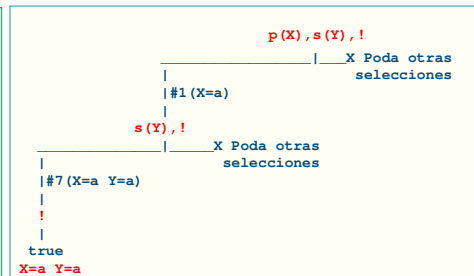
- Donde se podan las selecciones #2 y #3 de r(X), pero no las de s(Y).



- Y una más:

?- p(X), s(Y), !.
X=a Y=a ;
No

- Donde se podan tanto las selecciones #2 y #3 de r(X), como las selecciones #8 y #9 de s(Y), como se esperaba.



Ejemplo de aplicación

- En Prolog no hay `case_of` ni `if_then_else`
 - ◆ ...aunque se puede simular (algo parecido) así:

```
r(X) {  
  case x of  
    1: write(1);  
    2: write(2);  
    otherwise: write(3)  
}
```

```
r(X) :- X == 1, !, write(1).  
r(X) :- X == 2, !, write(2).  
r(_) :- write(3).
```

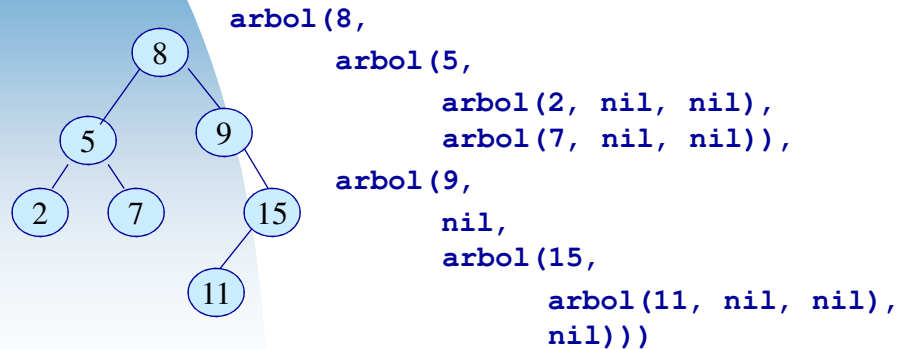
```
maximo(X,Y,Z) {  
  if X > Y then Z:=X  
  else Z:=Y  
}
```

```
maximo(X,Y,X) :- X > Y, !.  
maximo(_,Y,Y).
```

Estructuras de datos en Prolog

- Se puede utilizar la representación basada en listas (como en Scheme), pero ...
- también se puede aprovechar la representación por medio de funciones:
- EJEMPLO: **Árbol binario**
`arbol(raiz,`
 `subarbol_izquierdo,`
 `subarbol_derecho)`

Ejemplo



Recorrido en PREORDEN

```
preorden(arbol(R, I, D), X) :-  
  !, preorden(I, Rizq),  
  preorden(D, Rder),  
  append([R|Rizq], Rder, X).  
  
preorden(nil, []).
```

Verificar si un dato es miembro de un árbol

```
miembro(X, arbol(X, _, _)) :- !.
```

```
miembro(X, arbol(_, Izq, _)) :-  
    miembro(X, Izq), !.
```

```
miembro(X, arbol(_, _, Der)) :-  
    miembro(X, Der).
```

Escritura formateada

- El predicado `write()` imprime su argumento (uno solo):

```
write('Hello world').  
myhello(X) :- write('Hello '),  
              write(X), write('!').
```

- Los saltos de línea son con “`nl`”:

```
write('una linea'),nl,  
write('y otra linea'),nl.
```

Lectura de datos

- Prolog puede leer datos de archivos o de la terminal mediante el predicado `read(X)`.
- **Ejemplo:** Desplegar el promedio de un alumno dado.

```
promread :-  
    write('matricula del alumno? '), nl,  
    read(Mat),  
    promalum(Mat, Prom),  
    calif(Mat, Nom, _), !,  
    write('El promedio del alumno '),  
    write(Nom), write(' es '),  
    write(Prom), nl.
```

Uso de la cláusula fail

- Hace fracasar una meta.
- Es útil para forzar al sistema a entregar todos los resultados.
- **Ejemplo:** Listar los promedios de todos los alumnos.

```
promalum :-  
    calif(_, Nom, par(Uno, Dos, Tres)),  
    Prom is (Uno+Dos+Tres)/3,  
    write('El promedio del alumno '),  
    write(Nom), write(' es '),  
    write(Prom), nl,  
    fail.
```