

Análisis del Código

1. solve(true).

true siempre se debe satisfacer. En particular, constituye el caso base del planteo recursivo.

2. solve((A, B)) :- solve(A), solve(B).

La meta conjuntiva (A, B) se satisface sólo si se satisface tanto A como B.

Análisis del Código

3. solve(A) :- clause(A, B), solve(B).

Si la meta A es la cabeza de una regla con cuerpo B, entonces satisfacer A es equivalente a satisfacer (recursivamente) a todas las metas del cuerpo B.

clause/2 es un predicado predefinido que determina si en el programa Prolog actual existe alguna regla con la cabeza y el cuerpo suministrados como argumentos.

Uso del predicado Clause/2

p(X) :- q(X). p(b) :- r(b), s(b).
q(a).

?- clause(p(a), Cuerpo).

Cuerpo = q(a)

?- clause(p(b), Cuerpo).

Cuerpo = q(b);

Cuerpo = r(b), s(b)

?- clause(q(a), Cuerpo).

Cuerpo = true

Aplicaciones

- Monitorear el progreso de la resolución de una consulta.
- Extender al lenguaje mediante nuevos predicados predefinidos.
- Modificar la semántica de alguna de las partes del lenguaje.
- No es gratis: se paga un costo en términos de eficiencia.

Ejemplo Concreto

Definir un metaintérprete que cuente la cantidad de hechos visitados a lo largo de la resolución de una cierta consulta:

solve(true, 1).

solve((A, B), CHAB) :-
 solve(A, CHA), solve(B, CHB),
 CHAB is CHA + CHB.

solve(A, CH) :-
 clause(A, B), solve(B, CH).