**Ejecución de Programas**

**¿Cómo funciona PROLOG?**

* En Prolog **NO** hay instrucciones de control (if, for, …)
* Su ejecución se basa en dos conceptos:
  + **Unificación**: cada meta determina un subconjunto de clausulas susceptibles de ser ejecutadas. (**Punto de Elección**)
  + **Backtracking**: Prolog selecciona el primer punto de elección y sigue ejecutando el programa hasta determinar si la meta es verdadera o falsa.

**Backtracking**

* En caso de que un punto de elección sea falso entra en juego el **backtracking.**
* Consiste en deshacer todo lo ejecutado situando el programa en el mismo estado en que el que estaba justo antes de llegar al punto de elección.
* Entonces se toma el siguiente punto de elección que estaba pendiente y se repite de nuevo el proceso.

**Condiciones y Cálculos**

**Operadores Relacionales**

Azúcar sintética: Conforman clausulas en formato infijo y por si mismos arrojan un valor de verdad.

* Igual que =:=
* Diferente que =\=
* Menor que <
* Mayor que >
* Menor igual que =<
* Mayor igual que >=

**Operadores Aritméticos**

Conforman clausulas en formato infijo cuyo resultado debe instanciarse a una variable a través del operador **is**.

* suma +



* resta -



* multiplicación \*



* división /



* residuo *mod*

**Operador is**

variable **is** expresión\_aritmetica

* Es requisito utilizarlo cuando se requiere hacer una evaluación aritmética.
* Instancia a la variable con el resultado de la expresión y la clausula es verdadera por default.

**Programación Recursiva**

**Mecanismos de control**

No existe la iteración en Prolog, pero se puede simular.

* Agregar un **fail,** se aplica el back tracking.
* P: - g(X), write(x), nl, fail.



* P.

**Función para simular un ciclo**

1. Generador de soluciones
2. Cuerpo del ciclo
3. Test (Para que dispare el back tracking)

**Alternativa: Reglas Recursivas**

Reglas cuyo cuerpo, tienen términos que corresponden a la cabeza de la propia regla.

**Pensamiento recursivo**

* Al menos una clausula para el caso base y al menos una regla recursiva.
* La decisión de evaluar un caso u otro esta implícito en el lenguaje.
* Los casos base van como hechos.

*Ejemplo Factorial de un numero*

factorial (0, 1).

factorial (N, R): -

X is N-1,



Factorial (X, W),

R is N\*W



**Errores comunes**

* Prolog no permite anidar operaciones
* Los cálculos de las variables se deben hacer antes o después, no en la llamada recursiva.
* Asignación destructiva, no se puede hacer X is X – 1, porque X es diferente a X – 1
  + Para solucionarlo se debe usar una variable diferente.

**Entradas y salidas**

Write

El predicado write () imprime su argumento, uno solo:

* Write (‘Hello’).
* myhello(X): - write(‘Hello’), write(X), write (‘!’).

Los saltos de línea son con el átomo de **nl**

**Lectura de datos**

Prolog puede leer datos de archivos o de la terminal mediante el predicado read(X).

**Uso de la clausula fail**

* Hace fracasar una meta.
* Siempre regresa el valor de falso.
* Su utilidad es forzar el sistema para entregar todos los resultados, por el back tracking.