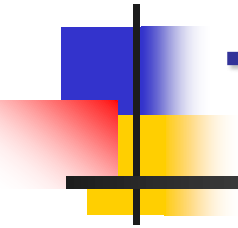

Tema 2. La Sintaxis

1. El vocabulario de un programa PROLOG
2. Términos
    - 2.1. Constantes
    - 2.2. Variables
    - 2.3. Estructuras
3. Operadores
4. Igualdad y Desigualdad
5. Aritmética en los programas PROLOG

1. El vocabulario de un programa PROLOG

- Sintaxis:
  - Conjunto de caracteres válidos (máx. 128; ASCII 7 bits):
    - Mayúsculas [A-Z]
    - Minúsculas [a-z]
    - Dígitos [0-9]
    - Caracteres especiales: + - * / ^ < > ~ : ; . _ ? \ ( ) [ ]
  - Se distinguen imprimibles de no imprimibles
  - Un programa PROLOG es una sucesión de términos:
    - Constantes
    - Variables
    - Estructuras (functores)

2. Términos

2.1. Constantes

- Expresan objetos :

  juan, maria,

- Hay dos clases de constantes:
  átomos y números
- Constantes átomo:
  - Constituidos por letras y números (y algunos signos)
    - comienzan por minúscula
    - pueden contener "_" (subrayado)
    - si van entre comillas simples (`), cualquier carácter
  - Constituidos por signos sólo:
    - ?- (asociado a una consulta)
    - :- (asociado a una regla)

Ejemplos

- Constantes válidas
  - a
  - vacio
  - 'jorge-perez'
  - -->
  - jorge_perez
  - algo1234
- Constantes no válidas
  - 2304algo
  - jorge-perez
  - Vacio
  - _alfa
- Constantes número:
  - Depende de la implementación
  - Al menos integer y float

2. Términos

2.2. Variables

- Representan objetos no específicos (a los que aún no sabemos cómo nombrar).
- Comienzan por Mayúscula ó "_" (subrayado)
  - X
  - Y
  - Z
  - Padre
  - Madre
  - Algo
  - _alfa
- "_" es una variable (anónima)
  - No queda ligada (no necesitaremos su valor)
  - ?- le_gusta_a (_, juan), le_gusta_a(_, luis)
  - Comparar con
    - ?- le_gusta_a (X, juan), le_gusta_a(X, luis)

2. Términos

2.3. **Estructuras** (Functor, Functores)

- Identifica un objeto compuesto (similar a un dato estructurado):
  colección de objetos o componentes
- Los componentes pueden ser a su vez estructuras
- Permite tratar como un **único objeto** a una **colección de informaciones relacionadas**
- También permite nombrar de forma diferente varios objetos de un mismo tipo

2.3. Estructuras

- Misma sintaxis que la de los hechos: *functor*(comp#1, comp#2,...).
- En general, *functor* es un nombre que designa:
  - un hecho,
  - una relación,
  - una función
- Internamente se guardan en forma arborescente.
- Ejemplo: fichas de libros de tu biblioteca
  - tiene(belarmino, libro).
  - tiene(belarmino, hobbit).
  - tiene(belarmino, el_senor_de_los_anillos).
  - tiene(belarmino, yo_robot).
  - tiene(belarmino, fundacion).

2.3. Estructuras (cont.)

Frente a:

- tiene(belarmino, libro(hobbit, tolkien)).
- tiene(belarmino, libro(el_senor_de_los_anillos, tolkien)).
- tiene(belarmino, libro(yo_robot, asimov)).
- tiene(belarmino, libro(fundacion, asimov)).

- tiene(belarmino, libro(hobbit, autor(jrr, tolkien))).
- tiene(belarmino, libro(yo_robot, autor(isaac, asimov))).

Ejercicio: Crea un fichero PROLOG, mi_biblioteca.pl, con una definición de tus libros, diferenciados por géneros, y después realiza consultas concretas, con variables y con variables anónimas.

3. Operadores

- Algunas estructuras/functores se pueden escribir como operadores:
  - x + y + z frente a +(z, +(x,y))
- Escribir un operador no implica evaluarlo.
  - ?- 4 = 2 + 2.
- Los operadores vendrán especificados por:
  - su posición (infija, prefija o postfija),
  - por su precedencia, y
  - por la asociatividad (por la izquierda o por la derecha).

3. Operadores (cont.)

- Trataremos los operadores aritméticos más habituales: +, -, *, y /.
- Los operadores aritméticos serán todos infijos.
- Las precedencias están asociadas a clases de operadores y varían según el compilador (entre 1 y 255):
  - los operadores * y / tendrán mayor precedencia siempre que + y –

3. Operadores (cont.)

- Los operadores aritméticos serán asociativos por la izquierda:
  - debe tener a su izquierda operaciones de precedencia menor o igual; a la derecha tendrá operadores de precedencia mayor
  - Ejemplos:
    - 8/2/2 será 2 u 8.
    - asociativo por la izquierda 8/2/2 = (8/2)/2

- Importante: los operadores aritméticos son como cualquier otra estructura:
  - pero pueden evaluarse con **is**
    - ?- X is 2 + 2.
    - ?- X is +(2,2).
    - ?- 4 is +(2,2).

4. Igualdad y Desigualdad

- El predicado *igualdad*, =, está predefinido:
  - *?- X = Y.*

  PROLOG para satisfacer el objetivo *comprueba si ambas pueden ligarse al mismo objeto*.

- Reglas para decidir si X e Y son "iguales":
  - Si X no está instanciada e Y sí, entonces son iguales y X toma como valor el término al que estuviese instanciado Y.
  - Los números y los átomos siempre son iguales entre sí.
  - Dos estructuras son iguales si tienen el mismo functor y el mismo número de argumentos, y cada uno de esos argumentos son iguales.

4. Igualdad y Desigualdad (cont.)

?- mesa = mesa.

?- silla = silla.

?- mesa = silla.

?- 2005 = 2004.

?- tiene(belarmino, X) = tiene(belarmino, libro(fundacion, asimov)).

?- tiene(belarmino, libro(_, X)) = tiene(belarmino, libro(yo_robot, author(isaac, asimov))).

- Si las 2 variables no están instanciadas, se cumple la igualdad y pasan a ser variables *compartidas*.
  - ?- X = Y.
    (Cuando una de ellas quede instanciada, fijará el valor de la otra.)

4. Igualdad y Desigualdad (cont.)

- Si las 2 variables no están instanciadas, se cumple la igualdad y pasan a ser variables *compartidas*.
  - ?- X = Y.
    (Cuando una de ellas quede instanciada, fijará el valor de la otra.)

  No tiene interés de forma aislada, pero:
  - ?- X=Y, X is 2+2, Z = Y.

4. Igualdad y Desigualdad (cont.)

- Existe otro predicado predefinido: *no es igual,* \=
  - ?- X \= Y.
    Se satisface si no se cumple el objetivo X = Y.

- Al igual que =, al estar predefinido no podemos modificarlos:

?- queso = yogurt.
?- mesa \= silla.
?- X = 4, Y = 2+2, X\= Y.

Ejercicio: Comprueba si se cumplen:

libro(titulo, autor) = libro(yo_robot, asimov).
libro(Titulo, Autor) = libro(yo_robot, asimov).
'constante' = constante.
fun1(a, b) = fun1(A, B).
fun1(a, b, c) = fun1(A, B).

5. Aritmética en los programas PROLOG

- Los programas PROLOG no están pensados para requerir una gran manipulación numérica
- Operadores comparación (predefinidos):
  - X =:= Y
  - X \= Y
  - X < Y
  - X > Y
  - X >= Y
  - X =< Y
- Operadores aritméticos:
  - X + Y
  - X – Y
  - X * Y
  - X / Y
  - X mod Y

5. Aritmética en los programas PROLOG (cont.)

- Los operadores aritméticos pueden usarse para hacer cálculos si se combinan con el operador **is**:
  - operador infijo,
  - el argumento a la derecha debe asociarse a un operador aritmético,
  - obliga a evaluar el argumento de la derecha y contrasta el resultado con el argumento a la izquierda:
    - X is 2 + 2.
    - resto is X mod Y.
  - No todos los operadores pueden evaluarse como expresiones aritméticas.