

## Introducción al cálculo proposicional

En la actividad anterior, se han evaluado (manualmente y automáticamente) expresiones como

- h)  $2 * (3+1) > 4$
- i)  $\forall n \in \{x \in \mathbb{N}, x < 100\}, n \bmod 10 \neq 9$
- j)  $\exists n \in \{x \in \mathbb{N}, x < 100\}, n \bmod 10 \neq 9$
- m) “abc” < “abd”

y se han obtenido valores del conjunto Bool, esto es true o false.

Las expresiones que contienen cuantificadores ( $\forall$  y  $\exists$ ), como la i) y la j) arriba, pertenecen al cálculo de predicados, que veremos más adelante. Las expresiones como la h) y la m) pertenecen al cálculo proposicional, son proposiciones.

Algunas expresiones no son proposiciones ni predicados, por ejemplo

- k)  $y \bmod 10 = 9$

Para esta expresión no se puede obtener un valor, porque contiene una *variable libre*, la  $y$ , de la cual no conocemos ni su valor ni su tipo.

¿Cómo podemos cambiar la expresión k) para poder obtener un valor de la misma?

Las proposiciones y los predicados no contienen variables libres. Las variables que no son libres, se llaman *variables ligadas*.

¿Cómo definiría ud. variable libre y variable ligada?

En esta unidad trabajamos con proposiciones.

Considere la siguiente secuencia de expresiones.

```
A := {1,2,3};  
x := 2;  
p := x in A;  
x := 5;  
p;  
x;  
B := {2,4,6};  
q := not (x in B);
```

$p = q;$

Evalúelas con ISetL:

Responda las siguientes preguntas:

1. ¿Qué significa el símbolo  $:=$ ?
2. ¿Y el símbolo  $=$ ?
3. ¿Qué símbolos se usan en matemática para uno y otro significado?
4. ¿Qué valor tiene  $x$  en  $p$  y en  $q$ ?

**Observe:** que una expresión como  $\text{not}(x \in B)$  es una proposición en este contexto, pues las variables  $x$  y  $B$  están ligadas por las definiciones anteriores que les asignan un valor. Para cada sesión de trabajo debe haber un contexto de referencia.

1. ¿Qué significa la palabra **not**?
2. ¿Conoce otros operadores del cálculo proposicional? Indique qué significa cada uno de ellos.
3. Ingrese en ISetL el siguiente segmento de código, píntelo y clique el ícono “semáforo verde” de la barra del menú:

```
for p, q in [true, false] do
    write p, q, p and q;
    writeln;
end;
```

¿Qué despliega el programa?

**Observe:** que a las variables  $p$  y  $q$  se les asigna cada uno de los valores `true` y `false`.

4. Escriba segmentos de código similares para los otros operadores que ud. ha indicado, así como para el operador `not`.
5. El operador `xor` (exclusive or) es true si y solo si **sólo uno de sus dos operandos** es true, false si no. Implemente un segmento de código para la tabla de verdad del operador `xor`, usando la siguiente equivalencia lógica:  $p \text{ xor } q = (p \text{ and } (\text{not } q)) \text{ or } ((\text{not } p) \text{ and } q)$ .

## **Algunas funciones predefinidas en ISetL**

Una expresión tiene un valor y un tipo, su expresión canónica es un natural, o un entero, o un conjunto, o una función, o una n-upla, o un string o un valor booleano, etc.

En ISetL, existen funciones predefinidas de la forma **is\_palabra** donde **palabra** puede ser **integer**, **func**, **tuple**, **set**, **string**, **is\_number**, etc. Estas funciones, aplicadas a un argumento, devuelven true si el mismo es del tipo que denota la palabra, false si no. Por ejemplo, **is\_integer** (2) es true mientras que **is\_integer** ("a") es false.

Practique en ISetL el uso de dichas funciones.

Otras funciones predefinidas en ISetL son:

**div**, (a div b es el cociente de la división entera entre los enteros a y b)  
**mod**, (a mod b es el resto de la división entera entre los enteros a y b)  
**min**, (min(a,b) es el menor de los enteros a y b o de los strings a y b)  
**max**, (max(a,b) es el mayor de los enteros a y b o de los strings a y b)  
**abs**, (abs(a) es el valor absoluto del entero a)  
**even**, (even(a) es true si a es par, false si no)  
**odd**, (odd(a) es true si a es impar, false si no)  
**union**, (A union B es la unión de los conjuntos A y B )  
**inter**, (A inter B es la intersección de los conjuntos A y B )  
**subset**, (A subset B es true si A es subconjunto de B, false si no)  
**pow**, (pow(A) es el conjunto de todos los subconjuntos del conjunto A)

Practique en ISetL el uso de cada una de estas funciones.