

Definiciones

- 1) Introduzca y evalúe las siguientes definiciones y expresiones en ISetL

```
x:=2;  
A:={0,2,4,6};  
p:= x in A;  
y:= x+5;  
A = {};  
p:= y in A;
```

Observe: que si ud. define una variable más de una vez, la que queda accesible es la última (caso de p arriba).

Proposiciones

- 1) Implemente en ISetL segmentos de código para obtener las tablas de verdad de las siguientes proposiciones:

```
not p and (p impl q)  
r and (s or not p)  
(not q) impl p = p or q
```

- 2) Una proposición tal que su tabla de verdad contiene solo el valor true para cualquier valor de las variables involucradas se llama tautología. Compruebe que ese es el caso de la última proposición de arriba.

- 3) Escriba una proposición que sea una contradicción (la tabla de verdad solo contiene el valor false para cualquier valor de sus variables) e implemente en ISetL su tabla de verdad.

- 4) Compruebe que $p \text{ impl } q$ es true si p es false, cualquiera sea el valor de q

Más elementos para trabajar en ISetL

A continuación se describen las cláusulas if y la definición de conjuntos por comprensión en ISetL. En ambos casos se utilizan expresiones booleanas.

Cláusulas if

Una cláusula if es un segmento de código que usan (casi) todos los lenguajes de programación y que generalmente puede ser de dos formas:

Si p entonces a
Si p entonces a si no b

donde p es una expresión booleana y a y b son listas de expresiones cualesquiera (expresiones separadas por ;)

En ISetL se escriben respectivamente así:

1. if p then a; end;
2. if p then a else b; end;

El efecto de ejecutar una cláusula if en ISetL es:

Caso 1: Si p evalúa a true, se evalúan cada una de las expresiones de la lista a. Si p evalúa a false, el resultado es indefinido.

Caso 2: Si p evalúa a true, se evalúan cada una de las expresiones de la lista a. Si p evalúa a false, se evalúan cada una de las expresiones de la lista b.

Para que el resultado de la evaluación de las expresiones de las listas a y b sea *visible*, es necesario usar alguna de las funciones *print*, *write* o *writeln*, como se muestra en los siguientes ejemplos:

```
if 2 < 3 then print(2 in {1,2,3}); print (odd(2));end;
if 2 < 3 then print(3 mod 2) else if even (3) then print(3) else print (3-1);
end;end;
x:=-4;
if is_integer(x) and (x < 0) then print(abs(x)) else if x >= 0 then print(x-1);
end;end;
```

Si no se usa la función *print*, la expresión se ejecuta pero el resultado no se muestra en pantalla.

Observe: que es necesario escribir tantos “end;” como “if” y que antes del else no va punto y coma.

Ejercicios

- 1) Defina las siguientes variables proposicionales:
 - a. p como “2 es un entero y es par”,
 - b. q como “el cociente de la división de 33 por 2 es impar”,
 - c. r como “el mínimo de 2 y 3 es mayor que 1”,
 - d. s como “el máximo de 2 y 3 es igual 3”
- 2) Escriba y evalúe en ISetL las siguientes proposiciones:
 - e. Si p entonces si $(2 - 1) > 0$ entonces imprima $2 - 1$.
 - f. Si q entonces imprima el resto de esa división si no imprima ambos dividendo y divisor
 - g. Si r o s entonces imprima el valor de “ $2 \notin \{-1,0,1\}$ ” si no imprima el valor de “3 es impar y es natural”.

Definición de conjuntos por comprensión en Isetl

Las definiciones de conjuntos por comprensión usadas comúnmente en matemática, se pueden escribir en ISetL de la siguiente manera:

{ exp(x, y, ...) : x in X, y in Y, ... | P(x,y, ...) };

donde X, Y, ... deben ser un conjuntos finitos, P(x,y, ...) es una expresión booleana y exp(x, y, ...) es cualquier expresión válida del lenguaje. P(x, y, ...) puede no indicarse.

Ejemplos:

Sean los conjuntos $A := \{-1,0,1\}$ y $S := \{1..3\}$.

Los siguientes son conjuntos definidos por comprensión en ISetL:

```
B:= {even(x) : x in S};  
C:= {x : x in S | even(x)};  
D:= {x : x in (A union S)};  
E:= {X : X in pow(A) | forall x in X | even(x)};  
F:= {e : e in {0..10} | (e mod 3 = 0) and (e /= 0)};  
G:= {[x,y] : x in A, y in S};  
H:= {[x,y] : [x,y] in G | x = 0};
```

Observe: que los pares ordenados se escriben en ISetL usando paréntesis rectos. G es el producto cartesiano de A y S, mientras que H es una relación binaria entre A y S. La notación se generaliza para las n-uplas, de modo que por ejemplo [“a”, true, 1] es la terna formada por esos elementos (en ese orden).

Ejercicios:

1. Escriba las correspondientes definiciones por comprensión de cada uno de los conjuntos B, C, D, E, F, G y H del ejemplo arriba en matemática.
2. Escriba por extensión cada uno de ellos.
3. Introduzca las definiciones en ISetL y compare sus respuestas.