5. Lógica proposicional: Conectivas y tablas de verdad

Ejercicios resueltos

■ Ejercicio 1.

Definir una conectiva cuya tabla de verdad sea:

p	q	$p \leftarrow q$
V	V	V
V	F	V
F	V	F
F	F	V

Solución:

1) Asignamos a la nueva conectiva la función Implinversa[] para poder usarla en lo sucesivo cuando nos aparezca de nuevo. Para definir esta función en primer lugar podemos usar la orden Implies[]:

```
Implinversa[p_, q_] := Implies[q, p]
```

Comprobamos que su tabla de verdad coincide con la dada:

```
Implinversa[True, True]
Implinversa[True, False]
Implinversa[False, True]
Implinversa[False, False]
```

```
True
```

```
True
```

```
False
```

```
True
```

2) Utilizando la orden condicional Which[], podemos definir una nueva función Implinversa2[] mediante:

```
Implinversa2[p_, q_] := Which[p == False && q == True, False, True, True]
```

Notesé como de nuevo tiene la misma tabla de verdad que la dada en el ejercicio:

```
Implinversa2[True, True]
Implinversa2[True, False]
Implinversa2[False, True]
Implinversa2[False, False]
```

```
True
```

```
True
```

```
False
```

```
True
```

3) Otra forma de definir esta conectiva sería usando la equivalencia lógica $(p \leftarrow q) \Leftrightarrow (\sim(q \wedge(\sim p)))$:

```
Implinversa3[p_, q_] := ! (q && (! p))
```

■ Ejercicio 2.

a) Escribir funciones que representen las conectivas: NOR y NAND.

b) Dadas las formas enunciativas:

b.1) A = [(p → q) ∧ s] → [(~r ← p) → q]

b.2) B = (p ↔ q) ↓ (r ↑ s)

b.3) C = (((~p) → s) ← (((~r) → (~q)) → p)).

Evaluarlas para p = Verdad, q = Verdad, r = Falso y s = Verdad.

c) Calcular las tablas de verdad de las formas enunciativas A, B y C.

Solución:

a) Definimos las conectivas NOR y NAND utilizando la definición de dichas conectivas, es decir, negando la disyunción y la conjunción respectivamente.

```
NOR[p_, q_] := ! (p || q)
```

```
— General::spell1 : Possible spelling error: new
    symbol name "NOR" is similar to existing symbol "Nor". More…
```

El mensaje de error que nos muestra *Mathematica* no es exactamente un error, sino que nos avisa de la existencia de una función muy parecida a la que hemos definido. De hecho, las conectivas NOR y NAND ya están definidas en *Mathematica* con el nombre Nor[] y Nand[], respectivamente.

```
NAND[p_, q_] := ! (p && q)
```

b) Asignamos los nombres AA, BB y CC a las tres formas enunciativas con las que tenemos que trabajar, escribiendo previamente todos las conectivas que *Mathematica* no tiene definidas y que vamos a utilizar:

```
Implinversa[p_, q_] := Implies[q, p]
NOR[p_, q_] := ! (p || q)
NAND[p_, q_] := ! (p && q)
Sii[a_, b_] := Implies[a, b] && Implies[b, a]
```

```
AA := Implies[Implies[p, q] && s, Implies[Implinversa[! r, p], q]]
BB := NOR[Sii[p, q], NAND[r, s]]
CC := Implinversa[Implies[! p, s], Implies[! r, Implies[! q, p]]]
```

Asignamos a las variables de enunciado los valores para los cuales se nos pide que evaluemos las tres formas enunciativas anteriores:

```
p = True;
q = True;
r = False;
s = True;
```

Evaluamos AA para los valores anteriores usando la orden TrueQ[]

```
TrueQ[AA]
```

```
True
```

Ahora para evaluar las dos siguiente sólo escribiremos el nombre que les hemos asignado:

```
BB
```

```
False
```

```
CC
```

```
True
```

c) Calculamos la tabla de verdad para la forma enunciativa AA cambiando en el programa que nos calcula las tablas de verdad el número de variables "n" y "expresion":

```
Implinversa[p_, q_] := Implies[q, p];
n = 4;
p = Table[False, {t, n}];
expresion := Implies[Implies[p[[1]], p[[2]]] && p[[4]],
   Implies[Implinversa[! p[[3]], p[[1]]], p[[2]]]];
tabla = Table["F", {x, (2^n + 1)}, {y, n + 1}];
For[k = 1, k < n + 1, k++, tabla[[1, k]] = \!\("p" \_ k\)];
tabla[[1, n + 1]] = "Exp";
For[i = 0, i < 2^n, i++,
    j = i;
    For[f = n, f > 0, f--,
        resto = Mod[j, 2];
        j = Floor[j / 2];
        If[resto == 0,
    p[[f]] = True; tabla[[i + 2, f]] = "V", p[[f]] = False];
         ];
    If[TrueQ[expresion], tabla[[i + 2, n + 1]] = "V"];
    ];
TableForm[tabla]
```

p_1	p_2	p_3	p_4	Exp
V	V	V	V	V
V	V	V	F	V
V	V	F	V	V
V	V	F	F	V
V	F	V	V	V
V	F	V	F	V
V	F	F	V	V
V	F	F	F	V
F	V	V	V	V
F	V	V	F	V
F	V	F	V	V
F	V	F	F	V
F	F	V	V	F
F	F	V	F	V
F	F	F	V	F
F	F	F	F	V

La tabla de verdad de B vendrá dada por:

```
Sii[a_, b_] := Implies[a, b] && Implies[b, a];
NOR[p_, q_] := ! (p || q);
NAND[p_, q_] := ! (p && q);
n = 4;
p = Table[False, {t, n}];
expresion := NOR[Sii[p[[1]], p[[2]]], NAND[p[[3]], p[[4]]]];
tabla = Table["F", {x, (2^n + 1)}, {y, n + 1}];
For[k = 1, k < n + 1, k++, tabla[[1, k]] = \!\("p" \_ k\)];
tabla[[1, n + 1]] = "Exp";
For[i = 0, i < 2^n, i++,
    j = i;
    For[f = n, f > 0, f--,
        resto = Mod[j, 2];
        j = Floor[j / 2];
        If[resto == 0,
    p[[f]] = True; tabla[[i + 2, f]] = "V", p[[f]] = False];
        ];
    If[TrueQ[expresion], tabla[[i + 2, n + 1]] = "V"];
    ];
TableForm[tabla]
```

p_1	p_2	p_3	p_4	Exp
V	V	V	V	F
V	V	V	F	F
V	V	F	V	F
V	V	F	F	F
V	F	V	V	V
V	F	V	F	F
V	F	F	V	F
V	F	F	F	F
F	V	V	V	V
F	V	V	F	F
F	V	F	V	F
F	V	F	F	F
F	F	V	V	F
F	F	V	F	F
F	F	F	V	F
F	F	F	F	F

De nuevo, introducciendo primero las conectivas que *Mathematica* no tiene definidas y que necesitamos, obtenemos la tabla de verdad de C mediante:

```
Implinversa[p_, q_] := Implies[q, p];
n = 4;
p = Table[False, {t, n}];
expresion := Implinversa[Implies[! p[[1]], p[[4]]],
   Implies[! p[[3]], Implies[! p[[2]], p[[1]]]]];
tabla = Table["F", {x, (2^n + 1)}, {y, n + 1}];
For[k = 1, k < n + 1, k++, tabla[[1, k]] = \!\("p" \_ k\)];
tabla[[1, n + 1]] = "Exp";
For[i = 0, i < 2^n, i++,
    j = i;
    For[f = n, f > 0, f--,
        resto = Mod[j, 2];
        j = Floor[j / 2];
        If[resto == 0,
    p[[f]] = True; tabla[[i + 2, f]] = "V", p[[f]] = False];
        ];
    If[TrueQ[expresion], tabla[[i + 2, n + 1]] = "V"];
    ];
TableForm[tabla]
```

p_1	p_2	p_3	p_4	Exp
V	V	V	V	V
V	V	V	F	V
V	V	F	V	V
V	V	F	F	V
V	F	V	V	V
V	F	V	F	V
V	F	F	V	V
V	F	F	F	V
F	V	V	V	V
F	V	V	F	F
F	V	F	V	V
F	V	F	F	F
F	F	V	V	V
F	F	V	F	F
F	F	F	V	V
F	F	F	F	V