PRÁCTICA Nº5

Lógica con el Mathematica (PARTE II)

Tautologías, contradicciones, formas normales equivalencias e implicaciones lógicas y argumentaciones.

1.- INTRODUCCIÓN.

Haciendo pequeñas variaciones sobre el programa que calculaba tablas de verdad de la práctica anterior obtendremos otros programas que deduzcan si una forma enunciativa es tautología o contradicción, que calculen las formas normales, si dos formas enunciativas son equivalentes, o si una forma argumentativa es válida.

2.- TAUTOLOGÍAS Y CONTRADICCIONES.

Como ya disponemos de un programa que nos permite calcular la tabla de verdad de una forma enunciativa cualquiera, comprobar si es tautología o contradicción una forma enunciativa debe de ser bastante sencillo. El procedimiento que vamos a seguir consistirá en utilizar parte del programa anterior, fijándonos sólo en si aparece algún caso que sea falso para rechazar la hipótesis de que sea tautología, o verdadero para contradicción.

Ejemplo 1:

Comprobar si la forma enunciativa: $(p_1 \vee p_2) \wedge (p_3 \vee p_4)$ es tautología o contradicción.

SOLUCIÓN:

Para comprobar si es tautología utilizamos el programa:

PROGRAMA	COMENTARIOS
n=4;	Nº de variables de enunciado.

<table>
<tr><td>tautologia=True;</td><td colspan="2">Por defecto supondremos que es tautología</td></tr>
<tr><td>p=Table[False,{t,n}];</td><td colspan="2">Definimos las variables de enunciado</td></tr>
<tr><td>expresion:=(p[[1]] || p[[2]]) && (p[[3]] || p[[4]]);</td><td colspan="2">Forma enunciativa que queremos comprobar si es tautología.</td></tr>
<tr><td>For[i=0,i<2^n,i++,
j=i;</td><td></td><td rowspan="4">Bucle que recorre todas las posibles combinaciones en la tabla de verdad.</td></tr>
<tr><td>For[f=n,f>0,f--,
resto=Mod[j,2];
j=Floor[j/2];
If[resto==0,p[[f]]=True,p[[f]]=False];
];</td><td>Bucle que calcula la combinación en cada caso.</td></tr>
<tr><td>If[TrueQ[expresion],Null,tautologia=False];</td><td>Comprueba si el valor de expresion en algún caso es falso.</td></tr>
<tr><td>];</td><td></td></tr>
<tr><td>tautologia</td><td colspan="2">Nos dice si es o no tautología.</td></tr>
</table>

Para comprobar si es contradicción utilizamos el programa:

<table>
<tr><th>PROGRAMA</th><th colspan="2">COMENTARIOS</th></tr>
<tr><td>n=4;</td><td colspan="2">Nº de variables de enunciado.</td></tr>
<tr><td>contradiccion=True;</td><td colspan="2">Por defecto supondremos que es contradicción</td></tr>
<tr><td>p=Table[False,{t,n}];</td><td colspan="2">Definimos las variables de enunciado</td></tr>
<tr><td>Expresion:=(p[[1]] || p[[2]]) && (p[[3]] || p[[4]]);</td><td colspan="2">Forma enunciativa que queremos comprobar si es contradicción.</td></tr>
<tr><td>For[i=0,i<2^n,i++,
j=i;</td><td></td><td rowspan="4">Bucle que recorre todas las posibles combinaciones en la tabla de verdad.</td></tr>
<tr><td>For[f=n,f>0,f--,
resto=Mod[j,2];
j=Floor[j/2];
If[resto==0,p[[f]]=True,p[[f]]=False];
];</td><td>Bucle que calcula la combinación en cada caso.</td></tr>
<tr><td>If[TrueQ[expresion],contradiccion=False];</td><td>Comprueba si el valor de expresion en algún caso es verdadero.</td></tr>
<tr><td>];</td><td></td></tr>
<tr><td>contradiccion</td><td colspan="2">Nos dice si es o no contradicción.</td></tr>
</table>

Ejercicio 1:

Comprobar si son tautologías o contradicciones las siguientes formas enunciativas:

a) $((p \wedge (\sim q)) \vee ((q \wedge (\sim r)) \vee (r \wedge (\sim p))))$;
b) $(p \leftrightarrow (\sim(\sim p)))$;
c) $(((\sim p) \rightarrow q) \rightarrow (((\sim p) \rightarrow (\sim q)) \rightarrow p))$;
d) $(p \wedge (\sim p))$.

Solución:

3.- FORMAS NORMALES.

Como es conocido, determinar las formas normales de una forma enunciativa a partir de su tabla es bastante fácil.

Ejemplo 2.-

Calcular las formas normales de la forma enunciativa:

$$(p_1 \vee p_2) \rightarrow p_3$$

<u>SOLUCIÓN:</u>

<table>
<tr><th>PROGRAMA</th><th>COMENTARIOS</th></tr>
<tr><td>n=3;</td><td>Nº de variables de enunciado.</td></tr>
<tr><td>cadena="";
cadena2="";
cad="";
cad2="";</td><td></td></tr>
<tr><td>contradiccion=True;</td><td>Suponemos que es contradicción.</td></tr>
<tr><td>tautologia=True;</td><td>Suponemos que es tautología.</td></tr>
<tr><td>p=Table[False,{t,n}];</td><td>Definimos las variables de enunciado</td></tr>
<tr><td>expresion:=Implies[(p[[1]] || p[[2]]), p[[3]]];</td><td>Forma enunciativa a la que queremos calcularle las formas normales.</td></tr>
<tr><td><pre>For[i=0,i<2^n,i++,
     j=i;
     cad="";
     cad2="";</pre></td><td>Bucle que recorre todas las posibles combinaciones en la tabla de verdad.</td></tr>
<tr><td><pre>   For[f=n,f>0,f--,
          resto=Mod[j,2];
          j=Floor[j/2];
         If[resto==0,
              p[[f]]=True;
              If[f==n,
                     cad=StringJoin["p",ToString[f],cad],
                    cad=StringJoin["p",ToString[f]," ^ ",cad]];
               If[f==n,
                    cad2=StringJoin["(~p",ToString[f],")",cad2],
                    cad2=StringJoin["(~p",ToString[f],")"," v ",cad2]];
               ,p[[f]]=False;
               If[f==n,
                     cad=StringJoin["(~p",ToString[f],")",cad],
                     cad=StringJoin["(~p",ToString[f],")"," ^ ",cad]];
               If[f==n,
                     cad2=StringJoin["p",ToString[f],cad2],
                     cad2=StringJoin["p",ToString[f]," v ",cad2]];
         ];
     ];</pre></td><td>Bucle que calcula la combinación en cada caso y la posible expresión que se añadirá a la forma normal disyuntiva o conjuntiva.</td></tr>
<tr><td><pre>   If[TrueQ[expresion],</pre></td><td></td></tr>
</table>

If[cadena=="",cadena=StringJoin[cadena, "(", cad, ")"], **cadena=StringJoin[cadena, " v (",cad,")"]];** **contradiccion=False;** **,** **If[cadena2=="",cadena2=StringJoin[cadena2, "(", cad2, ")"],** **cadena2=StringJoin[cadena2, " ^ (",cad2,")"]];** **tautologia=False;** **];** **];**	Determina si es contradicción y va calculando la forma normal disyuntiva. Determina si es tautología y va calculando la forma normal conjuntiva.
If[contradiccion, Print["Es una contradicción."], **Print["No es contradicción y la forma normal disyuntiva es: “,cadena]];** **If[tautologia, Print["Es una tautología."],** **Print["No es tautología y la forma normal conmutativa es: ",cadena2]];**	Salida de resultados.

Ejercicio 2:

Calcular las formas normales disyuntiva y conjuntiva de las siguientes formas enunciativas:

a) $((p \vee (\sim q)) \rightarrow (\sim r))$;
b) $((p \wedge r) \leftrightarrow (q \wedge r))$.

Solución:

4.- EQUIVALENCIAS LÓGICAS E IMPLICACIONES LÓGICAS.

Dadas dos formas enunciativas A_1 y A_2, sabemos que ambas son equivalentes si la forma enunciativa $A_1 \leftrightarrow A_2$ es tautología, por tanto utilizando el programa que determina cuando una forma enunciativa es tautología, tendremos un método para determinar cuando dos formas enunciativas son equivalentes.

Análogamente se razona para implicaciones lógicas.

Utilizando lo que acabamos de decir, resolver los siguientes ejercicios:

Ejercicio 3:

Comprobar que las siguientes formas enunciativas son equivalentes:

a) $(p \rightarrow q)$
b) $((\sim q) \rightarrow (\sim p))$

Solución:

Ejercicio 4:

Determinar cuales de estas formas enunciativas son equivalentes entre sí, y si alguna de ellas implica lógicamente a otra:

a) $((p \lor q) \land r)$;
b) $(((\sim p) \land (\sim q)) \to (\sim r))$;
c) $(r \to (q \land p))$;
d) $((p \land r) \lor (q \land r))$.

Solución:

5.- ARGUMENTACIONES VÁLIDAS.

Dada una argumentación cualquiera:

$$A_1, A_2, \ldots\ldots, A_n;\ \therefore A$$

recordemos que era válida si

$$(A_1 \wedge A_2 \wedge \ldots\ldots \wedge A_n) \rightarrow A$$

es una tautología. En consecuencia, al igual que ocurre en el apartado anterior, tenemos una forma directa de determinar cuando una forma argumentativa es válida.

Ejercicio 5:

Determinar si la siguiente forma argumentativa es válida:

$$p, (p|(q|r));\ \therefore r$$

Solución: