**Visual Studio Code y Word (para Juan Dent Herrera)**

A continuación código copiado de Visual Studio Code

(\* Generar la forma normal disyuntiva de una proposición con variables \*)

(\* Ignacio Trejos Zelaya, 2021 \*)

(\* Código escrito con base en mi conocimiento y la explicación dada en Murillo, M.

   'Introducción a la matemática discreta 4ta ed', sección 1.5 Formas normales \*)

(\* Funciones auxiliares \*)

(\* generar\_elt\_conj: genera un elemento conjuntivo:

   dada una variable var y un valor booleano, generar la partícula positiva o negada \*)

(\* para fnd: las variables van positivas cuando el valor es verdadero \*)

fun generar\_elt\_conj (var, valor) =

        if valor then variable var else negacion (variable var)

;

(\* generar\_conjuncion\_de\_elementos: dada una lista de pares (variable, valor booleano) \*)

fun generar\_conjuncion\_de\_elementos [par]                         =

        generar\_elt\_conj par

|   generar\_conjuncion\_de\_elementos (par1 :: par2 :: resto\_pares) =

        conjuncion (generar\_elt\_conj par1, generar\_conjuncion\_de\_elementos (par2 :: resto\_pares))

;

(\* generar\_disyuncion\_de\_conjunciones:

   dada una lista de asociaciones entre variables y valores booleanos \*)

fun generar\_disyuncion\_de\_conjunciones [asociacion] =

        generar\_conjuncion\_de\_elementos asociacion

|   generar\_disyuncion\_de\_conjunciones (asoc1 :: asoc2 :: resto\_asocs) =

        disyuncion (generar\_conjuncion\_de\_elementos asoc1, generar\_disyuncion\_de\_conjunciones (asoc2 :: resto\_asocs))

;

fun fnd prop =

    let

        (\* variables de trabajo \*)

        val variables = vars prop

        val n = length variables

        val lista\_combinaciones\_booleanas = gen\_bools n

        (\* generar evaluaciones de la proposición. Equivale a una tabla de verdad \*)

        fun recorrer []                  = []  (\* fin de generación de filas \*)

        |   recorrer (fila :: mas\_filas) =

            let

                (\* establecer una asociación entre variables y una combinación de valores booleanos (fila) \*)

                val asociacion = as\_vals variables fila

                (\* esta asociación constituye un ambiente, o contexto, para evaluar la proposición prop \*)

                val evaluacion\_es\_verdadera = evalProp asociacion prop

            in

                if evaluacion\_es\_verdadera then

                    asociacion :: recorrer mas\_filas

                else

                    recorrer mas\_filas

            end

    in

        if n = 0 then (\* no hay variables \*)

            constante (evalProp [] prop) (\* la proposición no necesita contexto para ser evaluada \*)

        else (\* inicio case \*)

            ( case recorrer lista\_combinaciones\_booleanas of

                []           => constante false  (\* es una contradicción, ninguna fila evaluó a verdadero \*)

              | asociaciones => generar\_disyuncion\_de\_conjunciones  asociaciones

            ) (\* fin case \*)

    end

;

Aquí sigue el texto de Word