Defensa Práctica III - Junio 2010

Ejercicio 1.- Eliminar el primer nivel de una expresión.

- El ejercicio consiste en programar (al estilo de las prácticas semanales de laboratorio) una función con la siguiente cabecera:

function RemoveFirstLevel(X : Expr; var ec : TException) : Expr;

- En vuestros respectivos escritorios, disponéis de una carpeta denominada DefensaJunio2010. La función se encuentra declarada en la Unit Ejercicios.pas. En dicho fichero debéis resolver y guardar la solución al ejercicio.

- El cometido de la función debe ser el de **eliminar** por completo el nivel inmediato inferior a la raíz de la expresión X pasada, **devolviendo una nueva expresión** que carezca de dicho nivel. **Ejemplo:** *X={{a,b},Sen[x,y]}* ➔ *RemoveFirstLevel[X]={a,b,x,y}*.

- Las expresiones pasadas a la función, deberán cumplir las siguientes reglas, provocando en caso contrario los errores a continuación descritos:

    1. **La expresión pasada debe ser siempre una lista o una función**. En caso contrario, la función deberá devolver el número de error 1, acompañado del mensaje de error "La función sólo trabaja sobre listas y funciones.". Ejemplo:

        - ✓ RemoveFirstLevel[León] ➔ Error 1.

    2. **El primer nivel del árbol de la expresión inmediatamente inferior a la raíz, deberá estar formado al completo por listas y funciones exclusivamente**. En caso contrario, la función debe devolver como número de error 2 y como mensaje "En el primer nivel inferior a la raíz sólo pueden aparecer listas y funciones.". Ejemplos:

        - ✓ RemoveFirstLevel[{a,b,c}] ➔ Error 2.
        - ✓ RemoveFirstLevel[{a,b},Sen[x,y],ABC] ➔Error 2.

- En la carpeta que contiene el código fuente de la práctica, tenéis disponible una batería de prueba con los resultados esperados. A continuación se muestran algunos ejemplos de lo que debe devolver la función a desarrollar:

    - ✓ RemoveFirstLevel[{Sen[],{a,Cos[r],b,Tan[r],c,},{x,y,z}}] ➔ {a,Cos[r],b,Tan[r],c,x,y,z}
    - ✓ RemoveFirstLevel[Sen[{a,b,c},Cos[x,y],{{{e}}},{p,o}] ➔ Sen[a,b,c,x,y,{{e}},p,o]
    - ✓ RemoveFirstLevel[{{x,y,{g,y},u},Cos[a,b,Sen[r,s]]}] ➔ {x,y,{g,y},u,a,b,Sen[r,s]}

Notas:

- **No** pueden aparecer como parte de la solución final las funciones **ExprToStr y ParseExpr**.
- **Obligatorio** un correcto funcionamiento con las opciones de compilación –Cr y –gh.
- Tiempo para realizar el ejercicio: **75 minutos.**

```
function RemoveFirstLevel(X : Expr; var ec : TException) : Expr;

var
        Ki,Ri:TExprIt;
        error:boolean;
        z:Expr;

begin
if (x<>nil) then begin
        if (x^.head='Symbol') then begin  //Si X es un símbolo, devolvemos el error 1.
                        ec.nerror:=1;
                        ec.msg:='La funcion solo trabaja sobre listas y funciones';
                        RemoveFirstLevel:=nil; end
        else begin
                 //Si la expresión no es un símbolo, copiamos la raíz como expresión resultado.
                error:=false;
                RemoveFirstLevel:=AllocExpr(X^.head, X^.terminal);
                MoveToFirstSubExpr(X,Ki);

                While ((not error) and (IsAtNode(Ki))) do begin
                        z:=ExprAt(Ki);
                        if (Z^.head='Symbol') then begin
                             //Si algún hijo es símbolo, devolvemos el error 2.
                                        ec.nerror:=2;
                                        ec.msg:='En el primer nivel no puede haber simbolos';
                                        ReleaseExpr(RemoveFirstLevel);
                                        RemoveFirstLevel:=nil;
                                        error:=true; end
                        else begin
                         //Vamos recorriendo la expresión original, añadiendo a la solución todas las
                        // sub-expresiones que sean hijas de las que están en el primer nivel bajo la raíz,
                        // las cuales no se añaden a la solución.
                                        MoveToFirstSubExpr(z,Ri);
                                        While (IsAtNode(Ri)) do          begin
                                                AddSubExpr(DeepCopy(ExprAt(Ri)),RemoveFirstLevel);
                                                MoveToNext(Ri); end;
                                        MoveToNext(Ki);
                        end;
                end;
        end;
  end;
end;
```