



Práctica 3:

DERIVADA DE UNA EXPRESIÓN

OBJETIVO

- Manipulación y reescritura de términos.
- Utilización del corte para representar casos por defecto.

PLANTEAMIENTO DEL PROBLEMA

En esta práctica se pide la realización de un programa que halle la derivada de una expresión y simplifique posteriormente el resultado.

El predicado para hallar la derivada se llamará *deriv/3*. El primer parámetro es la expresión a derivar, el segundo con respecto a qué se deriva y el tercer parámetro es la expresión derivada. Cada cláusula del predicado corresponderá a una regla de derivación. Por ejemplo, para derivar un número negativo, la cláusula correspondiente será:

```
deriv(-A, X, -U) :-  
    deriv(A, X, U).
```

Escribir las cláusulas necesarias para derivar expresiones que contengan sumas, restas, multiplicaciones y exponenciaciones (^) con exponentes enteros.

Si se observa la expresión resultante que devuelve *deriv/3*, se ve que es necesario simplificarla bastante para que obtenga un aspecto más familiar. Para ello utilizaremos el predicado *simp/2*, que también deberá ser implementado.

Cada cláusula del predicado *simp/2* servirá para simplificar una operación. Para ello, los pasos a realizar serán, en primer lugar, simplificar los operandos, y posteriormente, simplificar la operación (por ejemplo, evaluándola si es posible).

La cláusula de *simp/2* correspondiente a la suma sería la siguiente:

```
simp(A+B, Exp) :-  
    simp(A, A2),  
    simp(B, B2),  
    op(A2+B2, Exp).
```

Donde *op/2* corresponde al predicado encargado de simplificar la operación. Un ejemplo de cláusula de *op/2* podría ser la siguiente:

```
op(A+0, A).
```

Esta cláusula simplifica la suma cuando el segundo operando es cero.

NO es necesario que la expresión resultante esté totalmente simplificada.

NOTA: En esta práctica hay algunos casos en los que es necesario representar casos por defecto. La única forma de representar correctamente estos casos es mediante el uso del corte.

DURACIÓN DE LA PRÁCTICA:

1 Sesión