

Generalización del programa de comprobación del mantenimiento de las identidades matemáticas en una máquina de cómputo

Técnicas Experimentales
Práctica de Laboratorio #9

19 de abril de 2013

Resumen

El objetivo de esta práctica es entregar un módulo escrito en Python. Con el programa se comprobará, de forma general, el mantenimiento de las identidades matemáticas en una máquina de cómputo.

1. Motivación y Objetivos

Se quiere realizar la comprobación del siguiente listado de identidades matemáticas:

1. $(ab)^3$ y a^3b^3
2. $\frac{a}{b}$ y $\frac{1}{\frac{b}{a}}$
3. e^{a+b} y $e^a e^b$
4. $\log a^b$ y $b \log a$
5. $a - b$ y $-(b - a)$
6. $(ab)^4$ y a^4b^4
7. $(a + b)^2$ y $a^2 + 2ab + b^2$
8. $(a + b)(a - b)$ y $a^2 - b^2$
9. $\log ab$ y $\log a + \log b$
10. ab y $e^{\log a + \log b}$
11. $\frac{1}{(\frac{1}{a} + \frac{1}{b})}$ y $\frac{ab}{(a+b)}$
12. $a(\sin^2 b + \cos^2 b)$ y a
13. $\sinh(a + b)$ y $\frac{(e^a e^b - e^{-a} e^{-b})}{2}$
14. $\tan(a + b)$ y $\frac{\sin(a+b)}{\cos(a+b)}$
15. $\sin(a + b)$ y $\sin a \cos b + \sin b \cos a$

Nótese que en la práctica anterior se implementó un programa simple para comprobar algunas de las identidades de la lista, pero si se quieren comprobar todas las del listado, es mejor ser capaz de automatizar el proceso y escribir un procedimiento más general que reciba como parámetros:

- Las dos expresiones a comparar, es decir, *expr1* y *expr2*.
- El intervalo donde se han de realizar las comprobaciones, es decir, *A* y *B*.
- El número de pruebas que se quieren realizar.

Utilizar la sentencia `from math import *` en el fichero del módulo a crear, para que las funciones matemáticas como *'exp'* y *'log'* estén definidas.

2. Ejercicios propuestos

Escriba una función

```
equal(expr1, expr2, A, B, n)
```

que compruebe si las expresiones matemáticas '*expr1*' y '*expr2*', dadas como cadenas y escritas en función de números '*a*' y '*b*' son exactamente iguales:

```
eval(expr1) == eval(expr2)
```

para '*n*' elecciones aleatorias de los números '*a*' y '*b*' en el intervalo que va desde '*A*' a '*B*'. Se ha de devolver el porcentaje de fallos.

La llamada a la función `eval(s)` sobre una cadena `s`, que contiene una expresión *Python* la convierte en código como si el contenido de la cadena hubiera sido escrito directamente sobre el código del programa. El resultado de la siguiente llamada a `eval` es un objeto de tipo `float` que contiene el número 21.1:

```
>>> x = 20
>>> r = eval('x + 1.1')
>>> r
21.1
>>> type(r)
<type float>
>>>
```

Hacer un módulo con la función `equal` que se solicita. Hacer un bloque de comprobaciones que alimente a la función `equal` con argumentos leídos desde la línea de comandos. Importar la función `equal` desde el módulo que se ha creado y comprobar el listado inicial de quince identidades matemáticas. Almacenar todas las expresiones en una lista de 2-tuplas, donde cada 2-tuplas contienen dos expresiones matemáticamente equivalentes como cadenas, las cuales se pueden pasar como parámetro a la función `eval`.

Escriba una salida bien formateada en forma de tabla con una pareja de expresiones equivalentes en cada fila seguida de su porcentaje de fallos.

3. Para saber más...

Realice pruebas con valores de $A = 0$ y $B = 1$ y también con $A = -1e + 7$ y $B = 1e + 7$.

Explique si se puede concluir que el porcentaje de fallos parece depender de la magnitud de los números a y b .

Referencias

[1] Tutorial de Python. <http://docs.python.org/2/tutorial/>