In [1]:

```
from sympy import *
```

In [2]:

```
def halleys_method(function, initial, precision):
    x = Symbol('x')
    iters = 0

    lfunction = lambdify(x, function, 'numpy')

    dfunction = function.diff(x)
    ldfunction = lambdify(x, dfunction, 'numpy')

    d2function = dfunction.diff(x)
    ld2function = lambdify(x, d2function, 'numpy')

    while abs(lfunction(initial)) > precision:
        t = - lfunction(initial)/ldfunction(initial)
        r = ld2function(initial)*t**2/ldfunction(initial)
        initial = initial + t**2/(t + 0.5*r)
        iters += 1

    residual = lfunction(initial)
    return {'root': initial, 'iterations': iters, 'residual': residual}
```

In [3]:

```
x = Symbol('x')
func = poly(x**3 + 6*x**2 + 9*x - 4).as_expr()
halleys_method(func, 3.6, 0.001)
```

Out[3]:

```
{'root': 0.3553270038206912,
 'iterations': 3,
 'residual': 0.00034933313787899323}
```