64-ARITM-decimales-periodicos

December 3, 2017

Queremos obtener la expresión como decimal de una fracción, y la forma más simple es

```
In [1]: def decimal0(n,d,k):
            return (n/d).n(digits=k)

In [2]: decimal0(1,7,100)

Out[2]: 0.1428571428571428571428571428571428571428571428571428571428571428571428571428571428571
```

Sin embargo, y aunque sea más complicado, queremos ir obteniendo las cifras una a una con el fin de mantener más control sobre la forma en la que van apareciendo:

```
In [3]: def decimal(n,d,N):
            digitos = "0123456789"
            decimal= "0."
            r = (n%d)/d
            for muda in xsrange(N):
                r *= 10
                k = floor(r)
                decimal += digitos[k]
                r -= k
            return decimal

In [4]: decimal(1,7,100)

Out[4]: '0.1428571428571428571428571428571428571428571428571428571428571428571428571428571428571
```

Calculadas 100 cifras decimales de la fracción $1/7$ vemos que el resultado parece periódico con período 142857. Ahora queremos separar los decimales no periódicos y el período:

```
In [5]: def f(r):
            return floor(10*r)

In [6]: def F(r):
            return 10*r-f(r)

In [7]: def periodo(r):
            '''Suponemos n<d,  n primo con d, y d primo con 10'''
            digitos = '0123456789'
```

```
            n = r.numerator()
            d = r.denominator()
            des = '['
            orbita = []
            while d*r not in orbita:
                orbita.append(d*r)
                des += digitos[f(r)]
                r = F(r)
            des += "]"
            return des,orbita
```

```
In [8]: periodo(1/7)
```

```
Out[8]: ('[142857]', [1, 3, 2, 6, 4, 5])
```

```
In [9]: def no_period(r):
            '''Suponemos n<d y MCD(d,10)>1'''
            digitos = '0123456789'
            n = r.numerator()
            d = r.denominator()
            des = ''
            while gcd(10,r.denominator()) > 1:
                des += digitos[f(r)]
                r = F(r)
            return des,r
```

```
In [10]: def desarrollo(n,d):
             desarr = ''
             p_ent = str(n//d)+'.'
             n,d = n%d,d
             r = (n%d)/d
             C,r = no_period(r)
             C1,orbita = periodo(r)
             return p_ent+C+C1,orbita
```

```
In [11]: print desarrollo(107,450)
```

```
('0.23[7]', [7])
```

```
In [12]: print desarrollo(5,7)
```

```
('0.[714285]', [5, 1, 3, 2, 6, 4])
```

```
In [13]: print desarrollo(135,247)
```

```
('0.[546558704453441295]', [135, 115, 162, 138, 145, 215, 174, 11, 110, 112, 132, 85, 109, 102
```

```
In [14]: print desarrollo(1350,247)

('5.[465587044534412955]', [115, 162, 138, 145, 215, 174, 11, 110, 112, 132, 85, 109, 102, 32,

In [15]: (4/20).numerator()

Out[15]: 1

In [16]: print desarrollo(135,247)

('0.[546558704453441295]', [135, 115, 162, 138, 145, 215, 174, 11, 110, 112, 132, 85, 109, 102

In [17]: print desarrollo(1350,247)

('5.[465587044534412955]', [115, 162, 138, 145, 215, 174, 11, 110, 112, 132, 85, 109, 102, 32,
```