

序列 (Sequence) 的sympy操作

[hawksoft](#)

序列 (Sequence) 的sympy操作

1. 基本概念
2. 建立序列
3. 序列运算
4. 序列项连加
5. 序列项连乘
6. 其它

1. 基本概念

序列 (sequence)，定义为一系列有序对象。这些对象称为序列的项 (term)。一个序列包含项的数目可以是有限的，也可以无限，但必须是可数的。因此，按照顺序，可以给序列的每一项给一个自然数的编号，称为索引。

你可以把序列理解为python中list。

序列有两种表示方式：

1. 罗列法。如 $a_0, a_1, a_2, \dots, a_n$
2. 生成公式法。序列可以看成是定义在自然数集合 N 到序列项的集合上的双射函数，因此可以通过函数来描述序列，即： $a_n = f(n)$ 。 $f(n)$ 称为序列的生成公式。

生成公式是表示和理解序列的基本方法。

另外要注意：两个序列即使生成公式相同，若其定义域不同（尽管都是 N 的子集），仍然是不同的序列。序列的定义域称为序列的区间 (interval)。

2 建立序列

记住：定义一个序列，既要定义其生成公式，还要定义它的区间。

在sympy中，类SeqFormula用来通过公式法生成一个序列对象。序列对象的常见属性和方法很容易理解，其中的gen和interval属性，分别对应序列的生成公式和区间。

```

1  #这段代码只是为了美化输出，定义了一个输出latex公式字符串的函数lprint
2  def lprint(content,First = False,Last = False):
3      if First:
4          print('上面代码的输出如下（每次调用输出一行）：\n\n')
5          print('$$')
6          print('\begin{align}')
7          s = latex(content)
8          #print('{} {} {}'.format('&&',s,'\\'))
9          print('{} {} {} {}'.format('&',s,'&', '\\'))
10         if Last:
11             print('\end{align}')
12             print('$$')

```

```

1  from sympy import SeqFormula, pprint, oo, latex
2  from sympy.abc import n, a, b, r
3  s1 = SeqFormula(a*r**n, (n, 0, 8)) #创建一个下标从0到8的几何序列
4  lprint(s1, First=True)             #显示序列
5  lprint(s1.start)                   #序列的开始和结束下标
6  lprint(s1.stop)
7  lprint(s1.length)                  #序列长度
8  lprint(s1.interval)
9
10 s2 = SeqFormula(a*r**n, (n, 3, 5)) #创建一个下标从3到5的几何序列
11 lprint(latex(s2))                  #显示序列
12 lprint(s2.start)                   #序列的开始和结束下标
13 lprint(s2.stop)
14 lprint(s2.length)                  #序列长度
15 lprint(s2.interval)
16
17 lprint(s1.gen)                     #序列的生成公式
18 lprint(s2.gen, Last=True)          #序列的生成公式

```

上面代码的输出如下（每次调用输出一行）：

$[a, ar, ar^2, ar^3, \dots]$	(1)
0	(2)
8	(3)
9	(4)
[0, 8]	(5)
$[ar^3, ar^4, ar^5]$	(6)
3	(7)
5	(8)
3	(9)
[3, 5]	(10)
ar^n	(11)
ar^n	(12)

有些具有周期性的序列，可以用类SeqPer来建立。

```
1 from sympy import SeqPer
2 s2 = SeqPer((1,2,3),(n,0,5))
3 lprint(s2[:],First=True)
4 lprint(s2.period)
5 lprint(s2.periodical)
6 lprint(s2.start)           #序列的开始和结束下标
7 lprint(s2.stop)
8 lprint(s2.length)         #序列长度
9 lprint(s2.interval,Last=True)
```

上面代码的输出如下（每次调用输出一行）：

[1, 2, 3, 1, 2, 3]	(13)
3	(14)
(1, 2, 3)	(15)
0	(16)
5	(17)
6	(18)
[0, 5]	(19)

3 序列运算

序列运算，是基于对应项进行运算。（sympy在实现中，分别定义了对应的符号类，如：SeqAdd, SeqMul等，现在可以不理解）。

```
1 from sympy import SeqFormula, oo
2 from sympy.abc import m,n,a,b,r
3 s1 = SeqFormula(a*r**n,(n,0,8))
4 lprint(s1,First=True)
5 s2 = SeqFormula(b*r**m,(m,0,8))
6 lprint(s2)
7 lprint(s1 + s2)
8 lprint(s1 - s2)
9 lprint(s1 * s2)
10 lprint(s1 / s2)
11 lprint(s1.coeff_mul(2),Last=True)
```

上面代码的输出如下（每次调用输出一行）：

$$[a, ar, ar^2, ar^3, \dots] \quad (20)$$

$$[b, br, br^2, br^3, \dots] \quad (21)$$

$$[a + b, ar + br, ar^2 + br^2, ar^3 + br^3, \dots] \quad (22)$$

$$[a - b, ar - br, ar^2 - br^2, ar^3 - br^3, \dots] \quad (23)$$

$$[ab, abr^2, abr^4, abr^6, \dots] \quad (24)$$

$$[ab, abr^2, abr^4, abr^6, \dots] \quad (25)$$

$$[2a, 2ar, 2ar^2, 2ar^3, \dots] \quad (26)$$

4 序列项连加

序列求和，是通过函数`summation()`实现的。该函数需要两个参数，第一个是序列的生成公式，第二个是求和的区间。

```
1 from sympy import SeqFormula, oo, summation, factor, expand
2 from sympy.abc import m, n, a, b, r, k
3 s1 = SeqFormula(a*r**n, (n, 0, 8))
4 s11 = summation(s1.gen, (n, 0, 8))
5 lprint(s11, First=True)
6 lprint(factor(s11))
7 s12 = s11.subs([(a, 1), (r, 2)])
8 lprint(s12, Last=True)
```

上面代码的输出如下（每次调用输出一行）：

$$a(r^8 + r^7 + r^6 + r^5 + r^4 + r^3 + r^2 + r + 1) \quad (27)$$

$$a(r^2 + r + 1)(r^6 + r^3 + 1) \quad (28)$$

$$511 \quad (29)$$

注意符号运算与数值运算的区别。只有把 a 和 r 都指定了具体的数值，才能计算出数值结果来，其它都是符号结果。在sympy中，有时甚至所有的符号都指定了数值，仍然默认不会输出数值结果。如阶乘计算等。这可以执行一个方法`doit()`，才会输出数值结果。

```
1 from sympy import SeqFormula, oo, summation, factor, expand
2 from sympy.abc import m, n, a, b, r, k
3 s1 = SeqFormula(a*r**n, (n, 0, 8))
4 s11 = summation(s1.gen, (n, 0, m))
5 lprint(s11, First=True)
6 s12 = s11.subs([(a, 1), (r, 2)])
7 lprint(s12)
8 lprint(s12.subs(m, 10), Last=True)
```

上面代码的输出如下（每次调用输出一行）：

$$a \left(\begin{cases} m+1 & \text{for } r=1 \\ \frac{1-r^{m+1}}{1-r} & \text{otherwise} \end{cases} \right) \quad (30)$$

$$2^{m+1} - 1 \quad (31)$$

$$2047 \quad (32)$$

等差序列求和：

```
1 from sympy import *
2 from sympy.abc import a,r,k,n,d
3 s1 = SeqFormula(a*d*k,(k,0,8))
4 s11 = summation(s1.gen,(k,0,n))
5 lprint(s11,First=True)
6 s2 = s11.subs([(a,1),(d,1)])
7 lprint(s2)
8 lprint(factor(s2),Last=True)
```

上面代码的输出如下（每次调用输出一行）：

$$ad \left(\frac{n^2}{2} + \frac{n}{2} \right) \quad (33)$$

$$\frac{n^2}{2} + \frac{n}{2} \quad (34)$$

$$\frac{n(n+1)}{2} \quad (35)$$

阶乘序列求和：

```
1 from sympy import *
2 from sympy.abc import a,r,k,n,d
3 s1 = SeqFormula(a*factorial(k),(k,0,9))
4 s11 = summation(s1.gen,(k,0,n))
5 lprint(s11,First=True)
6 s2 = s11.subs(a,1)
7 lprint((s2.subs(n,10)).doit(),Last=True)
```

上面代码的输出如下（每次调用输出一行）：

$$\sum_{k=0}^n ak! \quad (36)$$

$$4037914 \quad (37)$$

5 序列项连乘

序列连乘，是通过函数`product()`实现的。与序列连加一样，该函数也需要两个参数，第一个是序列的生成公式，第二个是连乘的区间。

```

1 from sympy import SeqFormula, oo, product, factor, expand
2 from sympy.abc import m, n, a, b, r, k
3 s1 = SeqFormula(a*r**n, (n, 0, 8))
4 s11 = product(s1.gen, (n, 0, m))
5 lprint(s11, First=True)
6 s12 = s11.subs([(a, 1), (r, 2), (m, 3)])
7 lprint(s12, Last=True)

```

上面代码的输出如下（每次调用输出一行）：

$$a^{m+1} r^{\frac{m^2}{2} + \frac{m}{2}} \quad (38)$$

$$64 \quad (39)$$

6 其它

这里强调一下，不要把级数（series）与序列混淆。在sympy中，级数特指函数的泰勒级数，如下面的代码：

```

1 from sympy import series, cos, summation
2 from sympy.abc import x, y, z
3 s1 = series(cos(x), x, 0, 3)
4 lprint(s1, First=True)
5 s1 = series(x**2, x, y, 3)
6 lprint(s1, Last=True)

```

上面代码的输出如下（每次调用输出一行）：

$$1 - \frac{x^2}{2} + O(x^3) \quad (42)$$

$$y^2 + 2y(x - y) + (x - y)^2 \quad (43)$$