

sympy中的集合操作

[hawksoft](#)

sympy中的集合操作

- 1 常见集合
- 2 集合运算
- 3 集合关系运算
- 4 生成集合
- 5 集合与元素
- 6 多说两句

该部分首先定义了一个集合类Sets（注意与pythonne内建的Set类不是一个），然后：

- 以类方法的方式，定义了常见的集合操作，如集合的交、并、补等等
- 以该类为基类，定义了其它的一些特殊集合类并生成对象，如：Reals, Integers等等

所以，只要记住几个特殊的集合对象名字，并学习集合类的方法，就可以完成大部分工作。

1 常见集合

在sympy中，这些集合被定义为一个对象（尽管名字的第一个字母大写），所以可以直接使用。

- EmptySet #空集
- UniversalSet #全集
- Integers #整数集合
- Reals #实数集合
- Naturals #自然数集合（不含0）
- Natureals0 #自然数结合（含0）

看下面的代码。注意:其中pprint函数是sympy提供的一个特殊输出函数，代表pretty print。该函数配合环境，会采用图形化方式输出数学符号,因此比较美观易读。

```
1 from sympy import Naturals,Naturals0,Integers,Interval,Reals,pprint
2 pprint(Naturals)
3 pprint(Naturals0)
4 pprint(Integers)
5 pprint(Reals)
6 pprint(Naturals.contains(0))
7 pprint(Naturals0.contains(0))
```

```

1  N
2  No
3  Z
4  R
5  False
6  True

```

另外，我们常用的元素有限的集合，是以**FiniteSet**这个类提供的，可以用它生成我们需要的集合实例。

```

1  from sympy import FiniteSet
2  s1 =FiniteSet(1,2,4)
3  s2 =FiniteSet("a","b")
4  s3 = FiniteSet(s1,s2,1)
5  pprint(s1)
6  pprint(s2)
7  pprint(s3)

```

```

1  {1, 2, 4}
2  {a, b}
3  {1, {a, b}, {1, 2, 4}}

```

还有一个定义在实数上的区间，也是我们常见的一个无限集合，这个是由**Interval**类定义的。

```

1  from sympy import Interval
2  s1 = Interval(1,3)      #闭区间
3  pprint(s1)
4  s2 = Interval(1,3,left_open = True,right_open = True)  #左右开区间
5  pprint(s2)

```

```

1  [1, 3]
2  (1, 3)

```

2 集合运算

记住：

- 按照运算的封闭性，集合运算的结果仍然是集合
- 集合运算是作为集合类（Sets）的方法提供的。

```

1  from sympy import FiniteSet
2  s1 =FiniteSet(1,2)
3  s2 =FiniteSet(1,3)
4  pprint(s1)
5
6  pprint(s1.union(s2)) #并运算
7  pprint(s1+s2)       #也是并运算，这是符号重载

```

```

8
9 pprint(s1.intersect(s2)) #交运算
10 pprint(s1.intersection(s2)) #符号重载
11
12 pprint(s1.complement(s2)) #补运算
13 pprint(s2 - s1) # 减法
14
15 pprint(s1.powerset()) # 求幂集
16
17 pprint(s1 * s2) #笛卡尔乘积

```

```

1 {1, 2}
2 {1, 2, 3}
3 {1, 2, 3}
4 {1}
5 {1}
6 {3}
7 {3}
8 {∅, {1}, {2}, {1, 2}}
9 {1, 2} × {1, 3}

```

3 集合关系运算

记住：

- 集合关系运算是判断两个集合之间的关系，如子集，交集等,因此运算结果是一个bool类型的对象。
- 集合关系运算也是作为集合类（Sets）的方法提供的。

```

1 from sympy import FiniteSet as Set,Equality
2 s1 = Set(1,2)
3 s2 = Set(1,3)
4 s3 = Set(2,1)
5 pprint(s1.is_subset(s2)) #判断是否子集
6 pprint(s1.is_proper_subset(s2)) #判断是否真子集
7
8 pprint(s2.is_superset(s2)) #判断是否超集
9 pprint(s1.is_proper_superset(s2))
10
11 pprint(s1.is_disjoint(s2)) #判断两个集合是否不相交(即交集是否为空)
12
13
14 pprint(Equality(s1, s3)) #判断两个集合是否相等

```

```
1 False
2 False
3 True
4 False
5 False
6 True
```

4 生成集合

有时，我们需要从一个集合开始，按照某种规则，生成一个新的集合。sympy提供了两种方法。一种是从一个集合中，按条件选择一些元素，构成一个新集合，如下面的代码：

```
1 from sympy import FiniteSet, ConditionSet
2 from sympy.abc import x, y
3 s1 = FiniteSet(1, 2, 9, 4, 5, 6)
4 s2 = ConditionSet(x, x >= 4, s1)
5 pprint(s2)
```

```
1 {4, 5, 6, 9}
```

另一种是根据一个集合的元素，进行函数运算，生成一个新集合，如下面的代码：

```
1 from sympy import FiniteSet, ImageSet, Lambda
2 from sympy.abc import x, y
3 s1 = Set(1, 2)
4 s2 = ImageSet(Lambda(x, x*2), s1)
5 pprint(s2)
```

```
1 {2*x | x ∈ {1, 2}}
```

5 集合与元素

在编程时，我们经常会遇到集合与其成员关系的问题。对此：

1. 在sympy中，集合类有一个方法contains,用于判定元素是不是集合的成员。
2. 每一个集合都是一个可遍历对象，即可以采用python的for语句来访问它的每一个成员

看下面的代码：

```
1 from sympy import FiniteSet
2 s1 = FiniteSet(2, 3, 9, 4, 5)
3 pprint(s1.contains(9))
4 for element in s1:
5     print(element)
```

1	True
2	2
3	3
4	4
5	5
6	9

6 多说两句

其实集合类Sets还有几个方法或属性，是与集合的开、闭以及边界有关，大家现在可以不了解，但这些东西在学习泛函时就变得非常必要了。