Prerequisites

- sets (notation)

Sets with counters: Multisets

Sets are an abstract mathematical object that we may think of as collections in a lose sense. They are sometimes likened to bags, but this can be misleading. While bags convey the intuition that sets are unordered, they are unlike sets in that idempotency does not hold: a bag containing two apples is not the same thing as a bag containing one apple. Bags thus are closer to what mathematicians call **multisets**. Like sets, multisets are unordered, but idempotency does not hold; a multiset can contain multiple instances of the same object.

Definition 1. A **multiset** is a set where each element has a numerosity. We indicate that a set S is a multiset by prefixing it with a subscripted M, as in S_M. For all $s \in S_M$, $S_M(s)$ indicates the numerosity or **count** of s in S_M. We may write $S_M(s) = 0$ instead of $s \notin S_M$.

Example 1 The multiset $\{a, a, b\}_M$ contains two occurrences of a and one occurrence of b. Hence $\{a, a, b\}_M \neq \{a, b\}_M$ even though $\{a, a, b\} = \{a, b\}$. However, it still holds that order does not matter: $\{a, a, b\}_M = \{a, b, a\}_M = \{b, a, a\}_M$, just like $\{a, b\} = \{b, a\}$.

Exercise 1 Fill in the gaps with $=$ and $\neq$ as appropriate.

- $\{5, 5, 7, 8\}_M \_ \{7, 5, 8, 7\}_M$
- $\{5, 3, 4\} \_ \{5, 3, 4, 4, 3, 5, 5, 4, 3\}$
- $\{\text{peanut butter}, \text{jelly}\}_M \_ \{\text{peanut butter}, \text{jelly}\}_M$
- $\{\text{John}, \text{John}, \text{John}\}_M \_ \{\text{John}\}_M$
- $\{a\}_M \_ \{a, a\}$

The notation for multisets is much less standardized than that for sets, and not everybody follows the convention of subscripting multisets with M. If an author uses multisets, pay close attention to how they define their notation. Also, it is often convenient to explicitly list the count of each element rather than listing the element multiple times.

Example 2 The multiset $A_M := \{a, a, a, b, b, c\}$ can be more conveniently written as $A_M := \{a : 3, b : 2, c : 1\}$. Elements with a count of 0 are usually omitted, but may be included if this is relevant information.

```
from collections import Counter

def set_equals(A, B):
    print("{} same set as {}?".format(A,B), set(A) == (B))
```

```
def multiset_equals(A, B):
    print("{} same multiset as {}?".format(A,B), Counter(A) == Counter(B))

multiset_equals(["a", "a", "b", "b", "c"], ["a", "b", "c", "c", "d"])
set_equals(["a", "a", "b", "b", "c"], ["a", "b", "c", "c", "d"])
multiset_equals(["a", "a", "b", "b", "c"], ["a", "b", "c", "a", "b"])
```

Exercise 2 Represent all the multisets in the exercise above with explicit counts.

With this kind of notation, it also becomes possible to define multisets with set-builder notation. For example, $\{n : 2n \mid n \in \mathbb{N}\}$ is the set that contains $2n$ occurrences of every natural number n: $\{0 : 0, 1 : 2, 2 : 4, 3 : 6, \ldots\}$.

```
multiset = Counter({n: 2*n for n in range(10)})
print(multiset)
```

Exercise 3 Write down the multiset defined by each set-builder expression. These are not entirely straight-forward, and you'll have to make some educated guesses as to how to handle special cases.

- $\{n : 10 - n \mid 0 \leq n \leq 10\}$
- $\{a : b, b : a \mid a, b \geq 0, a + b = 10\}$