```
    self.test_for_collision()

def test_for_completeness(self):
    """
    Check all the keys in self.out.ruleset.reference_dict against the whole set
    of possibilities. Make sure that all possible cases are covered by one of
    the reference keys in self.out.ruleset.reference_dict.
    """
    key_space = list(itertools.product(*self.out.ruleset.dimensions))
    reference_keys = self.out.ruleset.reference_dict.keys()

    for reference_key in reference_keys:
        # Each reference key is a tuple. The elements of the tuple are integers
        # or tuples of integers. Wrap all the integer elements inside tuples,
        # so all the elements of prepared_ref_key are iterable.
        prepared_ref_key = itertools.imap(self.wrapScalarValueAndSort,
                                          reference_key,
                                         )

        # Take the distilled reference key and explode it in to keys which are
        # tuples of integers.
        for exploded_key in itertools.product(*prepared_ref_key):
            # If we have overlap in the keys, this will catch it.
            base_key = self.out.ruleset[exploded_key]
```