CSC 591: Software Testing | HW: 1

a. Describe the input domain for the intersection() method.

Solution: The input domain for the `intersection()` method consists of two sets, `s1` and `s2`, which can take the following values:

- `null`
- An empty set: `{}`
- A set with at least one element: `{...}`

b. Does the partition "Type of s1" satisfy the completeness property? If not, give a value for s1 that does not fit in any block.

Solution: Yes, the partition "Type of s1" satisfies the completeness property. The blocks cover all possible states for `s1`: `null`, empty set or has at least one element

c. Does the partition "Type of s1" satisfy the disjointness property? If not, give a value for s1 that fits in more than one block.

Solution: Yes, the partition "Type of s1" satisfies the disjointness property. There is no overlap between the blocks, as `null`, an empty set, and a non-empty set are mutually exclusive.

d. Does the partition "Relation between s1 and s2" satisfy the completeness property? If not, give a pair of values for s1 and s2 that does not fit in any block.

Solution: The partition "Relation between `s1` and `s2`" **does not** fully satisfy the completeness property because it does not cover the scenario where `s1` and `s2` have some overlapping elements but are not subsets of each other and are not disjoint.

For example: `s1 = {9, 10, 11}` and `s2 = {11, 12, 13}` is such a case that does not fit any of the provided blocks. As there is a partial overlap between the sets, but neither is a subset of the other.

e. Does the partition "Relation between s1 and s2" satisfy the disjointness property? If not, give a pair of values that fits in more than one block.

Solution: The partition "Relation between `s1` and `s2`" **does not** satisfy the disjointness property because the same pair of sets (`s1 = {3, 4}` and `s2 = {3, 4}`) fits into blocks 1, 2 and 3. This indicates that the blocks are not mutually exclusive, and the partition does not fully satisfy the disjointness property.

CSC 591: Software Testing | HW: 1

f. Change the blocks for the partitions "Type of s1" and "Relation between s1 and s2" such that they do not suffer from any disjointness or completeness problems, but do not just use partitions with two blocks (i.e., blocks that are effectively "true" and "false").

Solution: We can adjust the blocks as follows:

- **Type of s1**:
  - Block 1: `s1 = null`
  - Block 2: `s1 is an empty set`
  - Block 3: s1 has at least one element
- **Relation between s1 and s2**:
  - Block 1: `s1` and `s2` are equal
  - Block 2: `s1` is a proper subset of `s2`
  - Block 3: `s2` is a proper subset of `s1`
  - Block 4: `s1` and `s2` have no common elements
  - Block 5: `s1` and `s2` have some common elements, but neither is a subset of the other

g. Create a partition for Characteristic 2, "Type of s2" which is analogous to "Type of s1".

Solution: Type of s2:

- Block 1: `s2 = null`
- Block 2: `s2 is an empty set`
- Block 3: `s2 has` has at least one element

h. Choose a representative input for each block from the three partitions "Type of s1", "Type of s2", and "Relation between s1 and s2".

Solution:

Type of s1	Type of s2	Relation between s1 and s2
`s1 = null`	`s2 = null`	**Block1:** `s1 = {1, 2}, s2 = {1, 2}`
`s1 = {}`	`s2 = {}`	**Block2:** `s1 = {1}, s2 = {1, 2}`
`s1 = {1, 2}`	`s2 = {1}`	**Block3:** `s1 = {1, 2}, s2 = {1}`
-	-	**Block4:** `s1 = {1, 2}, s2 = {3, 4}`
-	-	**Block5:** `s1 = {1, 2}, s2 = {2, 3}`

CSC 591: Software Testing | HW: 1

i. Describe the constraints among the three partitions used in (h) above.

Solution:

Type of `s1 or s2`: `s1 or s2` can either be an empty set `{}` (or `null`, treated the same), or a set with at least one element (e.g., `{1, 2}`).

Relation between `s1` and `s2`:

- If both `s1` and `s2` are `{}` (or `null`), they are considered equal.
- If one set is `{}` (or `null`) and the other has elements, the empty set is a proper subset of the non-empty set.
- If both `s1` and `s2` have elements, their relationship can be:
    - `s1` is a proper subset of `s2` (Block 2).
    - `s2` is a proper subset of `s1` (Block 3).
    - `s1` and `s2` are disjoint, having no common elements (Block 4).
    - `s1` and `s2` have some common elements, but neither is a proper subset of the other (Block 5).

j. Construct test cases that obey the constraints among the three partitions and satisfy the pair-wise coverage criteria.

Pair-Wise Coverage (PWC) Criterion : *A value from each block for each characteristic must be combined with a value from every block for all other characteristics.*

Each test case should have two input sets and an expected output.

Solution:

Test Case 1: Tests the intersection of two `null` sets, expecting an empty set as the result.

Test Case 2: Tests the intersection of two empty sets, expecting an empty set as the result.

Test Case 3: Tests the intersection of a `null` set with a non-empty set, expecting an empty set as the result.

Test Case 4: Tests the intersection of an empty set with a non-empty set, expecting an empty set as the result.

CSC 591: Software Testing | HW: 1

Test Case 5: Tests the intersection of two identical non-empty sets, expecting the set itself as the result.

Test Case 6: Tests the intersection of two sets where the first is a subset of the second, expecting the first set as the result.

Test Case 7: Tests the intersection of two sets where the second is a subset of the first, expecting the second set as the result.

Test Case 8: Tests the intersection of two sets with some common elements, but neither is a proper subset of the other, expecting the common element as the result.

Test Case 9: Tests the intersection of two disjoint sets, expecting an empty set as the result.