Coverage-Based Test Design in Action

CS 3250 Software Testing

Activity: Task 1

Test design (You have ~10 minutes to complete this task)

- Form a team of 8-10, each team gets two bags of candies
- Examine bag #1
- Imagine you are conducting a "candy testing" – Yes, imagine, **don't eat yet** .. You will execute your tests later
- Discuss in your team, use the worksheet

1. Come up with one coverage criterion to test the candy
   (example, C = taste one candy of each texture)
2. Develop a set of test requirements that satisfies your criterion
   (example, TR = {hard, soft}, where tr1 = hard, tr2 = soft)
3. Develop a set of test cases that satisfies your test requirements
   (example, T = {two sweet tarts, one sour patch}, where t1 = two sweet tarts, t2 = one sour patch)
   assuming two sweet tarts are consumed at once – what if one is consumed at a time?

Activity: Task 2

Coverage level of your tests (~5 minutes)

- Execute your tests against `bag #1`
  - You will now transform yourself into a "human-PUT"
  - For each test case, the "human-PUT"
    - Takes input (candy)
    - Performs a "consume" operation
  - Expected output: normal behavior, "human-PUT" does not crash

Due to the no eating in class policy, please **imagine** the "consume" operation while you are in class – you will do the actual consume operation outside of class

- Evaluate your tests, use the worksheet
  4. Record which test requirements are satisfied by your test set
  5. Compute the coverage level. Be sure to consider infeasible test requirements

Activity: Task 3

Coverage level of another team's tests (~5 minutes)

- Trade your test design (from task 1) with another team
- Execute another team's test cases against bag #2
  - You will now transform yourself into a "human-PUT"
  - For each test case, the "human-PUT"
    - Takes input (candy)
    - Performs a "consume" operation
  - Expected output: normal behavior, "human-PUT" does not crash

Due to the no eating in class policy, please **imagine** the "consume" operation while you are in class – you will do the actual consume operation outside of class

- Evaluate the tests, use the worksheet
  6. Record which test requirements are satisfied by another team's test set
  7. Compute the coverage level. Be sure to consider infeasible test requirements

Activity: Wrap-up Questions

8. Is your criterion appropriate? Justify, why?

9. Is there redundancy in your test set?

10. Coverage level: Given your criterion, consider the coverage levels of your test set and another team's test set. Are they different? What does it mean if one is higher than another?

11. Comparing criteria: Compare your coverage criterion and another team's criterion. Does one subsume another? (Is one a subset of another? Is it a many-to-one or one-to-many mapping? Is it a one-to-one mapping?)