Software Product Lines

Exercise 11: Product-Line Testing

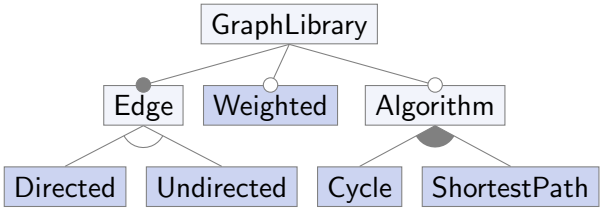
ISF (TU Braunschweig) January 2026

1. Combinatorial Interaction Testing

(a) How can pairwise interaction sampling be used to detect feature interactions in a software product line? What are the differences between pairwise and t-wise interaction sampling for $t > 2$? What is the difference between t-wise interaction coverage and statement, branch, or condition coverage?

(b) Determine all pairs of features ($t = 2$) relevant for sampling the following feature model. Which features and which pairs of features do not have to be considered?

Feature Model



Cycle $\rightarrow$ Directed

ShortestPath $\rightarrow$ Directed $\wedge$ Weighted

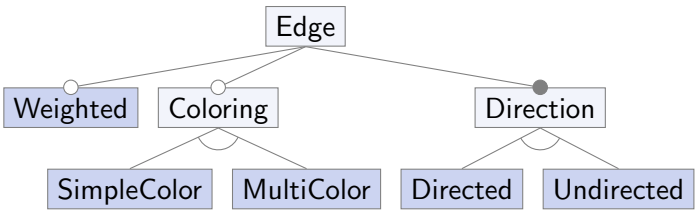
(c) Do the following configurations suffice to cover all relevant pairwise feature interactions from above? If they do not, add further configurations until all relevant pairwise interactions are covered. (For brevity, we abbreviate feature names to their first letter.)

Name	G	E	D	U	W	A	C	S
c_1	1	1	1	0	0	0	0	0
c_2	1	1	0	1	0	0	0	0
c_3	1	1	0	1	1	0	0	0
c_4	1	1	1	0	1	1	1	1

2. Solution-Space Sampling

Consider the following product line for a configurable `Edge` datatype:

Feature Model



MultiColor → Weighted

Annotated Source Code

```
class Edge {
  int cost = 1;
  //#if Weighted
  int weight = 3;
  //#endif
  //#if Coloring
  //#if SimpleColor
  int colorCost = 2;
  //#endif
  //#if MultiColor
  int[] colorCosts = {1, 2};
  //#endif
  //#endif

  int computeCost() {
    int total = cost;
    //#if Weighted
    total += weight;
    //#endif
    //...
    // ... (continued)

    //#if Coloring
    //#if SimpleColor
    total += colorCost;
    //#endif
    //#if MultiColor
    for (int c : colorCosts) {
      total += c;
    }
    //#endif
    //#endif
    //#if Directed
    return total * 2;
    //#elif Undirected
    return total;
    //#endif
  }
}
```

(a) Based on the above feature model and source code, construct a minimal sample of valid configurations that achieves coverage of `#if` blocks (Tartler et al. 2012). Justify your answer.

(b) Construct a second minimal sample that achieves pairwise presence-condition coverage (Krieter et al. 2022). Justify your answer.

(c) For each of the following strategies, how many configurations do we have to test at least?

i. Product-based strategy

ii. Sample-based strategy using `#if` block coverage

iii. Sample-based strategy using pairwise presence-condition coverage

Under which circumstances would each strategy be preferred?

3. Limits of Sample-Based Testing

(a) What are the advantages and disadvantages of relying on a small sample of configurations instead of testing all valid products in a software product line?

(b) Besides sample-based testing (covering t-wise interactions, `#if` blocks, or presence-conditions) and product-based testing, which alternative testing strategies can you think of? When would you use which testing strategy?

(c) In practice, how can a suitable value of t for t-wise interaction sampling be chosen? Discuss with examples or scenarios where lower or higher values of t might be appropriate.

(d) What is the difference between t-wise interaction coverage and t-wise presence-condition coverage? Discuss when which strategy is more appropriate.