Piacente Cristian 866020

Homework H8 – The subsume relation

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca

Identify the subsume relations among the following structural adequacy criteria (consider the inline definition given below, and find additional information in Chapter 12 of the book):

- branch coverage: execute each branch (edge of the CFG)
- simple condition coverage: execute each simple condition with both true and false values
- compound condition coverage: execute each decision with each combination of truth values that participate in the evaluation

Exemplify your answers with reference to the following code and the corresponding control flow diagram, and use the code and the control flow for reasoning about the subsume relations. The code and the control flow graph are also available as Figure 12.1 and 12.2 in the book. For this exercise, assume that the compiler is **not** enforcing short-circuit evaluation of conditional expressions. For the pairs of criteria that you compare, provide an intuitive justification of the existence of the subsume relation (or a counterexample that shows that the relation does not exist) with reference to the example.

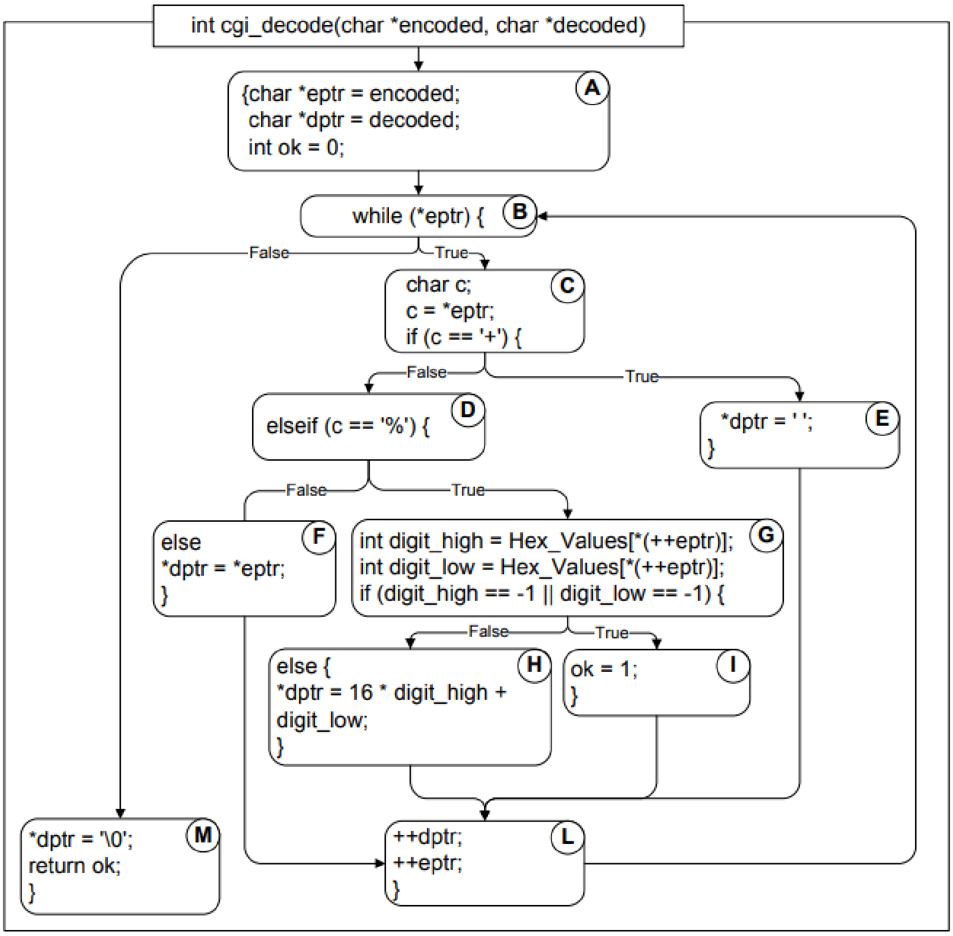
```
int cgi_decode(char *encoded, char *decoded) {
  char *eptr = encoded;
  char *dptr = decoded;
  int ok=0;
  while (*eptr) {
    char c;
    c = *eptr;
    if (c == '+') {
      *dptr = ' ‘;
    } else if (c == '%') {
      int  digit_high = Hex_Values[*(++eptr)];
      int  digit_low  = Hex_Values[*(++eptr)];
      if (digit_high == -1 || digit_low == -1) {
        ok=1;
      } else {
        *dptr = 16* digit_high + digit_low;
      }
    } else {
      *dptr = *eptr;
    }
    ++dptr;
    ++eptr;
  }
  *dptr = '\0';
  return ok;
}
```

Piacente Cristian 866020

Homework H8 – The subsume relation

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca

Let’s analyze the subsume relations among branch coverage, simple (also known as basic) condition coverage and compound condition coverage (assuming short-circuit evaluation is NOT enforced) w.r.t the cgi_decode function and its CFG shown below:



Branch coverage vs Basic condition coverage (and viceversa)

For branch coverage, we need to ensure that every branch is executed at least once, while for basic condition coverage each individual condition must evaluate to both true and false during testing.

There are test suites which satisfy the branch adequacy criterion but not the basic condition adequacy criterion. For instance, consider the following test suites:

T1 = {“adequate+test%0Dexecution%7U”}

T2 = {“ “, “+%0D+%4J”}

T1 and T2 cover all the branches without covering all the basic conditions: digit_high == -1 is never true. To satisfy the basic condition adequacy criterion, we would need to add a test case like “basic%K7”.

→ Branch coverage **does not subsume** basic condition coverage.

The opposite is also true: consider the test suite T3 = {“first+test%9Ktest%K9”}. The basic condition adequacy criterion is satisfied, but the basic block H, which would get executed when executing the False branch, is never executed.

→ Basic condition coverage **does not subsume** branch coverage, so they are not comparable.

Piacente Cristian 866020

Homework H8 – The subsume relation

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca

Basic condition coverage vs Compound condition coverage (and viceversa)

Compound condition coverage requires a test for each possible evaluation of compound conditions.
Since we are not taking in account short-circuit evaluation, we have 2^N combinations of N basic conditions (which is the worst case if short-circuit evaluation is considered).

Every test suite that satisfies the compound condition adequacy criterion also satisfies the basic condition adequacy criterion.

The justification is simple: if we test all combinations of basic conditions, then we inherently test each basic condition individually for both true and false values.

→ Compound condition coverage **does subsume** basic condition coverage.

The opposite is not true: testing each basic condition individually for both true and false values does not require to test all possible combinations of basic conditions.

→ Basic condition coverage **does not subsume** compound condition coverage.

Branch coverage vs Compound condition coverage (and viceversa)

There are test suites that satisfy the branch adequacy criterion, but not the compound condition adequacy criterion.

The justification is the following: branch coverage ensures that every possible branch is executed at least once, but it does not require that every possible combination of basic conditions is tested.

We have already seen that T1 and T2 cover all the branches and that we don’t have every basic condition evaluated to both true and false. This also shows that not every possible combination is tested, so compound condition adequacy criterion is not satisfied.

→ Branch coverage **does not subsume** compound condition coverage.

The opposite is true: if every possible combination of basic conditions is tested, we inherently include testing of each individual branch that could result from each of those combinations.

→ Compound condition coverage **does subsume** branch coverage.