Graph: Structural Coverage for Source Code

CS 3250 Software Testing

[Ammann and Offutt, “Introduction to Software Testing,” Ch. 7.3]
Structures for Criteria-Based Testing

Four structures for modeling software

- **Input space**
  - Graph
    - Source
    - Design
    - Specs
    - Use cases
  - Applied to: R--R
- **Logic**
  - Source
  - Specs
  - FSMs
  - DNF
  - Applied to: RI-R
- **Syntax**
  - Source
  - Models
  - Integration
  - Inputs
  - Applied to: RIPR
Overview

- Graph coverage criteria are widely used on source code
- Define graph, then apply coverage criterion
- **Control flow graph (CFG):** the most common graph for source code
- Node coverage: execute every statement
- Edge coverage: execute every branch
- Data flow coverage: augment the CFG with
  - **defs:** statements that assign values to variables
  - **uses:** statements that use variables
Control Flow Graph (CFG)

- Represent the control flow of a piece of source code

- **Nodes** represent basic blocks
  - **Basic blocks** represent sequences of instructions / statements that always execute together in sequence

- **Edges** represent control flow (branch) between basic blocks
  - Transfer of control

- **Initial nodes** correspond to a method’s entry points

- **Final nodes** correspond to a method’s exit points
  - Return or throw in Java

- **Decision nodes** represent choices in control flow
  - if or switch-case blocks or condition for loops in Java

- Can be annotated with extra information such as branch predicates, defs, and uses
Example: CFG for *if-else*

```plaintext
if (x < y) {
    y = 0;
    x = x+1;
} else {
    x = y;
}
```

- **Basic blocks (nodes)**
  1. if (x < y)
  2. y=0; x = x+1;
  3. x = y;
- **Entry node**
  1
- **Decision nodes**
  1
- **Junction nodes**
  4
- **Exit nodes**
  4

- **Control flow (edges)**
  1 → 2
  1 → 3
  2 → 4
  3 → 4

(Code snippet has been simplified)
Example: CFG for *If without else*

```c
if (x < y) {
    y = 0;
    x = x + 1;
}
```

- **Basic blocks (nodes)**
  1: `if (x < y)`
  2: `y = 0; x = x + 1;`

- **Entry node**
  1

- **Decision nodes**
  1

- **Junction nodes**
  3

- **Exit nodes**
  3

- **Control flow (edges)**
  1 → 2
  1 → 3
  2 → 3

(Code snippet has been simplified)
Example: CFG for *If* with *return*

- **Basic blocks (nodes)**
  1: if $(x < y)$
  2: return;
  3: print(x); return;

- **Entry node**
  1

- **Decision nodes**
  1

- **Junction nodes**
  -

- **Exit nodes**
  2, 3

- **Control flow (edges)**
  1 $\rightarrow$ 2
  1 $\rightarrow$ 3

(Code snippet has been simplified)
Loops

- Loops require **extra** nodes ("**dummy**" node)
  - Not directly derived from program statements

- Looping structures: **while** loop, **for** loop, **do-while** loop

- Common mistake
  - Try to have the edge go to the entry node
Example: CFG for a `while` loop

```plaintext
x = 0;
while (x < y)
{
    y = f(x,y);
    x = x + 1;
}
```

- **Basic blocks (nodes)**
  1: `x = 0;`
  2: `while(x < y)`
  3: `y = f(x,y); x = x+1;`

- **Control flow (edges)**
  1 → 2
  2 → 3
  2 → 4
  3 → 2

- **Entry node**
  1

- **Decision nodes**
  2

- **Junction nodes**
  -

- **Exit nodes**
  4

(Code snippet has been simplified)
Example: CFG for a for loop

```c
for (x=0; x<y; x++)
{
    y = f(x,y);
}
```

- Basic blocks (nodes)
  1: x = 0;
  2: x < y
  3: y = f(x,y);
  4: x++;

- Control flow (edges)
  1 → 2, 2 → 3, 2 → 5, 3 → 4, 4 → 2

- Entry node
  1

- Decision nodes
  2

- Junction nodes
  -

- Exit nodes
  4

(Code snippet has been simplified)
**Example: CFG for a do-while loop**

```
x = 0;
do {
    y = f(x, y);
    x = x + 1;
} while (x < y)
println(y);
```

**Basic blocks (nodes):**
1: `x = 0; do`
2: `y = f(x, y); x = x + 1; while(x < y)`
3: `println(y);`

**Control flow (edges):**
1 → 2
2 → 2
2 → 3

**Entry node:** 1
**Decision nodes:** 2
**Junction nodes:** -
**Exit nodes:** 3

(Code snippet has been simplified)
Example: CFG for a loop with `break` and `continue`

```python
x = 0;
while (x < y)
{
    y = f(x, y);
    if (y==0)
        break;
    else if (y < 0)
    {
        y = y*2;
        continue;
    }

    x = x + 1;
}
print(x);
```

(Code snippet has been simplified)
Example: CFG for *(switch) case*

```c
read(c);
switch(c)
{
    case 'N':
        y = 25;
        break;
    case 'Y':
        y = 50;
        break;
    default:
        y = 0;
        break;
}
```

Cases without break?
- Fall through to the next case

(CODE SNIPPET HAS BEEN SIMPLIFIED)
Example: CFG for Exceptions (try-catch)

```java
try {
    s = br.readLine();
    if (s.length() > 96)
        throw new Exception("too long");
    if (s.length() == 0)
        throw new Exception("too short");
} catch (IOException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.getMessage();
}
return(s);
```

(Code snippet has been simplified)
Summary

- A common application of graph coverage criteria is to program source – control flow graph (CFG)

- Applying graph coverage criteria to control flow graphs is relatively straightforward

- A few decisions must be made to translate control structures into the graph

- We use basic blocks when assigning program statements to nodes while some tools assign each statement to a unique node.
  - Coverage is the same, although the bookkeeping will differ