COMPSCI 702:
Software Measurement
McCabe’s Cyclomatic Complexity Number

Ewan Tempero  e.tempero@cs.auckland.ac.nz
www.cs.auckland.ac.nz:~ewan

Emilia Mendes (supervisor)  emilia@cs.auckland.ac.nz
Lecture Overview

- Structure complexity
- McCabe’s Cyclomatic Complexity Number
- References — Fenton and Pfleeger, Chapters 7 & 8
- Potential Exam Question
  Explain the intuition that supports using McCabe’s Cyclomatic Complexity Number as a metric for “complexity”.
Structure complexity: Intuition

• The more “complex” the structure of code
  ◦ … the harder it is to understand
  ◦ … the more likely it will have a defect
  ◦ … the harder it will be to change
  ◦ … the longer it will take to produce
  ◦ … the more difficult it will be to reuse
What is structure?

- control-flow — the sequence of instructions that are executed
- data-flow — the creation and movement of data between “components” of the code
- data organisation — the relationship of data items to each other
Directed Graphs

- a mathematical structure (with an appealing visual representation) for representing things that are related
- consists of *vertices* (or nodes, or points), connected by *edges* (or line segments, or arcs).
  - **Vertices:** A, B, C, D, E, F
  - **Edges:** (A,C), (A,C), (A,E), (B,C), (B,F), (C,D), (D,A), (E,F),

- Also: undirected graphs, rules restricting edges between vertices, classification of vertices
- *Graph Theory* — the study of properties of graphs
Flowgraphs

- Directed graphs (flowgraphs) can be used to model control flow of a program.
- \( vertex = \) statement, \( edge = (A, B) \) if control flows from statement \( A \) to \( B \).
- Properties of flowgraphs may provide information about properties of the code.
Example

public static boolean isPrime(int n) {
    boolean prime = true;
    int i = 2;
    while (i < n) {
        if (n % i == 0) {
            prime = false;
        }
        i++;
    }
    return prime;
}

McCabe’s Cyclomatic Complexity Number (CCN)

• measures the number of *linearly independent* paths through the flowgraph

• $v(F) = e - n + 2$, $F$ the flowgraph of the code, $n$ the number of vertices, $e$ the number of edges

• Intuition — the larger the CCN the “more complex” the code

• Various sources recommend a CCN of no more than 10-15

• Example: CCN = $8 - 7 + 2 = 3$
Example 2

```java
public static boolean isPrimeA(int n) {
    for (int i = 2; i < n; i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}
```
Example

if (a)
  if (b)
    goto d:
  else
    goto e:
else
  if (c)
    goto d:
  else
    goto e:
d: print "Hi."
e: print "How are you?"

\[
\text{CNN} = 11 - 9 + 2 = 4
\]

\[
\text{CNN} = 7 - 5 + 2 = 4
\]
From graph theory, $v(f) = d + 1$, where $d$ is the number of predicate nodes (tests/conditions)

```java
public static boolean isPrimeA(int n) {
    for (int i = 2; i < n; i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}

public static boolean dividableBy4(int n) {
    if (n % 2 == 0) {
        if (((n/2) % 2 == 0) {
            return true;
        }
    }
    return false;
}
```
Switch statements

```java
switch (a) {
    case one:
        break;
    case two: // fall through
    case three:
        break;
    default:
}

// do case one

if (a == one) {
    // do case one
}
else if (a == two) {
    // do case two
    // do case three
}
else if (a == three) {
    // do case three
}
else {
    // default
}
```
public void aMethod(Object obj) {
    // some code without conditionals
    obj.toString();
    // more code without conditionals
}

... try {
    // some code
    FileInputStream f = new FileInputStream("foo");
    // more code
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (SecurityException e2) {
    e2.printStackTrace();
} finally {
    System.out.println("finally!");
}...

Exceptions
Validation

- Attribute relation model: direct metric
- Unit Definition Model: number of linearly independent paths
- Instrumentation Model: construct the flowgraph of the code, determine the number of edges and vertices — compute or (much easier) identify condition vertices — compute
- Measurement protocol model: for loops, exceptions, switch statements
- Entity population Model: does it make sense to talk about “average” CCN?
- Representation Condition:
  - for complexity — dubious
  - for number of linearly independent paths — valid