Tutorial Outline

- What is White Box Testing?
- Flow Graph and Coverage Types
- Symbolic Execution:
  - Formal Definition
  - Examples
White-Box means Testing by Implementation

- Execution-based testing that uses the program’s inner structure and logical properties
  - A.K.A Clear Box, Glass Box and Structural Testing
- There are different types of white-box testing
  - For example statement coverage where each statement is executed at least once
- *Flow Graph* helps us model and analyze different types of coverage
Flow Graph

\( G = (V, E) \) where
- \( V \) is the set of basic blocks
- \( E \) is the set of control branches

Example:
1. \( a = \text{Read}(b) \)
2. \( c = 0 \)
3. while \( (a > 1) \)
4. \( \text{If } (a^2 > c) \)
5. \( c = c + a \)
6. \( a = a - 2 \)

Input: \( b = 2 \)

Output: \( a = 0, c = 2 \)
White Box Coverage Types

- **Statement** Coverage: Every statement is executed
- **Branch** Coverage: Every branch option is chosen
- **Path** Coverage: Every path is executed
- **Basic Path** Coverage:
  - We need to define basic path set first

Loops?
Basic Path Set

- An **execution path** is a set of nodes and directed edges in a flow graph that connects (in a directed fashion) the start node to a terminal node.
- Two execution paths are said to be **independent** if they do not include the same set of nodes and edges.
- A **basic** set of execution paths for a flow graph is an independent maximum set of paths in which all nodes and edges of the graph are included at least once.
Basic Path Coverage

- The number of Basic paths is \( E - N + 2 \) (Linear Complexity)

- Example
  
  p1 = start – 1,2 – 3 – end
  p2 = start – 1,2 – 3 – 4 – 6 – 3 – end
  p3 = start – 1,2 – 3 – 4 – 5 – 6 – 3 – end

  \[ E - N + 2 = 8 - 7 + 2 = 3 \]
Path Function

- A function \( f : D^n \rightarrow D^n \) represents the current values of the variables as function of their initial values.
- Each variable \( X \) is represented by a projection function \( f_X : D^n \rightarrow D \).
- Function composition \((g \circ f)(\overline{v}) = g(f_{x_1}(\overline{v}),...,f_{x_n}(\overline{v}))\)
  - For example

\[
\begin{align*}
f(X,Y,Z) &= (X + Y, X - Y, XZ) \\
f_x(X,Y,Z) &= X + Y \\
f_y(X,Y,Z) &= X - Y \\
f_z(X,Y,Z) &= XZ \\
 g(X,Y,Z) &= (XY, X + Z, Z) \\
(g \circ f)(X,Y,Z) &= g(f_x(X,Y,Z), f_y(X,Y,Z), f_z(X,Y,Z)) = \\
&= g(X + Y, X - Y, XZ) = ((X + Y)(X - Y), (X + Y) + XZ, XZ)
\end{align*}
\]
Path Condition

- A condition that ensures the execution of a path
- A constraint on the initial values of the variables

For Example: \( p = \text{start} - 1, 2 - 3 - \text{end} \).

1. \( a = \text{Read}(b) \)
2. \( c = 0 \)
3. while \( (a > 1) \)
4. if \( (a^2 > c) \)
5. \( c = c + a \)
6. \( a = a - 2 \)

The path condition is \( B \leq 1 \), where \( B \) is the initial value of \( b \).
Symbolic Execution

- A method for deriving test cases which satisfy a given path
  - Outputs path condition (input) and path function (expected result)
- Initially
  - Path function is the Identity function
  - Path condition is true
- Each step in the path induce a **symbolic composition** on the path function or a **logical constraint** on the path condition
  - Simple block $g(x)$: $f \leftarrow g \circ f$
  - Control branch: $C \leftarrow C \land \text{branch condition}$
Example: Symbolic Execution

1. $a = \text{Read}(b)$
2. $c = 0$
3. while ($a > 1$)
4. if ($a^2 > c$)
5. $c = c + a$
6. $a = a - 2$

Find test case for path:
$p = \text{start} - 1,2 - 3 - 4 - 5 - 6 - 3 - 4 - 5 - 6 - 3 - \text{end}$

White Box Testing
Example: Symbolic Execution

1. \( a = \text{Read}(b) \)
2. \( c = 0 \)
3. while \( (a > 1) \)
4. \( \quad \text{if } (a^2 > c) \)
5. \( \quad \; c = c + a \)
6. \( \quad \; a = a - 2 \)

\[
p = \text{start} - 1,2 - 3 - 4 - 5 - 6 - 3 - 4 - 5 - 6 - 3 - \text{end}
\]

<table>
<thead>
<tr>
<th>vertex</th>
<th>path function</th>
<th>path condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>start:</td>
<td>( (A, B, C) )</td>
<td>true</td>
</tr>
<tr>
<td>1,2</td>
<td>( (A, B, C) )</td>
<td>true</td>
</tr>
<tr>
<td>3</td>
<td>( (B, B, 0) )</td>
<td>true</td>
</tr>
<tr>
<td>4</td>
<td>( (B, B, 0) )</td>
<td>( (\text{true} \land B&gt;1) \leftrightarrow B&gt;1 )</td>
</tr>
<tr>
<td>5</td>
<td>( (B, B, 0) )</td>
<td>( (B&gt;1 \land B^2&gt;0) \leftrightarrow B&gt;1 )</td>
</tr>
</tbody>
</table>
Example: Symbolic Execution

1. \( a = \text{Read}(b) \)
2. \( c = 0 \)
3. while \((a > 1)\)
4. if \((a^2 > c)\)
5. \( c = c + a \)
6. \( a = a - 2 \)

\[
p = \text{start} - 1, 2 - 3 - 4 - 5 - 6 - 3 - 4 - 5 - 6 - 3 - \text{end}
\]

<table>
<thead>
<tr>
<th>vertex</th>
<th>path function</th>
<th>path condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>( B, B, B )</td>
<td>( B &gt; 1 )</td>
</tr>
<tr>
<td>3</td>
<td>( B-2, B, B )</td>
<td>( B &gt; 1 )</td>
</tr>
<tr>
<td>4</td>
<td>( B-2, B, B )</td>
<td>((B &gt; 1 \land B-2 &gt; 1) \leftrightarrow B &gt; 3)</td>
</tr>
<tr>
<td>5</td>
<td>( B-2, B, B )</td>
<td>((B &gt; 3 \land (B-2)^2 &gt; B) \leftrightarrow B &gt; 4)</td>
</tr>
<tr>
<td>6</td>
<td>( B-2, B, 2B-2 )</td>
<td>( B &gt; 4 )</td>
</tr>
<tr>
<td>3</td>
<td>( B-4, B, 2B-2 )</td>
<td>( B &gt; 4 )</td>
</tr>
</tbody>
</table>

end \( (B-4, B, 2B-2) \) \( (B > 4 \land B-4 \leq 1) \leftrightarrow B = 5 \)
Example: Symbolic Execution

1. \( a = \text{Read}(b) \)
2. \( c = 0 \)
3. while \( (a > 1) \)
4. if \( (a^2 > c) \)
5. \( c = c + a \)
6. \( a = a - 2 \)

\[ p = \text{start} - 1, 2 - 3 - 4 - 5 - 6 - 3 - 4 - 5 - 6 - 3 - \text{end} \]

\text{end} \quad (B-4, B, 2B-2) \quad B=5

Hence the test case is \( B = 5 \)
and the expected result is \( 2B-2 = 8 \)

Is there a test case for

\[ p = \text{start} - 1, 2 - 3 - 4 - 5 - 6 - 3 - 4 - 5 - 6 - 3 - \text{end} \]