**Dynamic** Backward Slicing

Input: A=2; B=3; C=3 Dynamic PDG

Void main

{

1. Cin >> a;
2. Cin >> b;
3. Cin >> c;
4. a = 1;
5. if (a<6)

{

1. c++;
2. b = c+b;

}

1. Else

{

1. c=5+c;

}

1. While ((B-A) >= c)

{

1. b=b+a;
2. c=c+b;

}

1. cout <<a;
2. cout <<b;
3. cout <<c;

}

1. Create the execution trace:
2. Create the Dynamic PDG
3. Compute a dynamic Backward Slice for
4. S(c, 15, last execution position) = ( )
5. S(A, 13, last execution position) = ( )

Question #2 **Dynamic** Forward Slicing

Given is the following input: A=2; B=3; C=3

Void main

{

1. Cin >> a;
2. Cin >> b;
3. Cin >> c;
4. a = 1;
5. if (a<6)

{

1. c++;
2. b = c+b;
3. If (B <C)

{

1. A++;

}

1. if ((B-A) >= c)

{

1. b=b+a;
2. A++

}

1. c=c+b;

}

1. cout <<a;
2. cout <<b;
3. cout <<c;

}

Solutions:

Input: A=2; B=3; C=3 Dynamic PDG

Void main

{

1. Cin >> a;
2. Cin >> b;
3. A diagram of numbers and circles

   Description automatically generatedCin >> c;
4. a = 1;
5. if (a<6)

{

1. c++;
2. b = c+b;

}

1. Else

{

1. c=5+c;

}

1. While ((B-A) >= c)

{

1. b=b+a;
2. c=c+b;

}

1. cout <<a;
2. cout <<b;
3. cout <<c;

}

1. Create the execution trace: 11, 22, 33, 44, 55, 66, 77, 108, 119, 1210, 1011, 1312, 1413, 1514
2. Create the Dynamic PDG
3. Compute a dynamic Backward Slice for
4. S(c, 15, last execution position) = ( 2,3,4,5, 6,7, 10, 11, 12, 15 )
5. S(A, 13, last execution position) = ( 4, 13 )

Question #2 Sample solution

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Void main  {   1. Cin >> a; 2. Cin >> b; 3. Cin >> c; 4. a = 1; 5. if (a<6)   {   1. c++; 2. b = c+b; 3. If (B <C)   {   1. A++;   }   1. if ((B-A) >= c)   {   1. b=b+a; 2. A++   }   1. c=c+b;   }   1. cout <<a; 2. cout <<b; 3. cout <<c;   } | |  |  |  |  | | --- | --- | --- | --- | | executed | A | B | C | |  |  |  |  | | X | 1 |  |  | | X | 1 | 2 |  | | X | 1 | 2 | 3 | | X | 4 | 2 | 3 | | X | 4,5 | 2,4,5 | 3,4,5 | |  |  |  |  | | X | 4,5 | 2,4,5 | 3,4,5,6 | | X | 4,5 | 2,3,4,5,6,7 | 3,4,5,6 | | X | 2,3,4,5,6,7,8 | 2,3,4,5,6,7,8 | 2,3,4,5,6,7,8 | |  |  |  |  | |  | 2,3,4,5,6,7,8 4,5 | 2,3,4,5,6,7,8 2,3,4,5,6,7 | 2,3,4,5,6,7,8 3,4,5,6 | | X | 2,3,4,5,6,7,10 | 2,3,4,5,6,7,10 | 2,3,4,5,6,7,10 | | X | 2,3,4,5,6,7,10 | 2,3,4,5,6,7,10,11 | 2,3,4,5,6,7,10 | | X | 2,3,4,5,6,7,10,12 | 2,3,4,5,6,7,10,11 | 2,3,4,5,6,7,10} | |  | 2,3,4,5,6,7,10,12 | 2,3,4,5,6,7,10,11 | 3,4,5,6 | | X | 2,3,4,5,6,7,10,12 | 2,3,4,5,6,7,10,11 | 2,3,4,5,6,7,10,11,13 | |  |  |  |  | | X | 2,3,4,5,6,7,10,12 | 2,3,4,5,6,7,10,11 | 2,3,4,5,6,7,10,11,13 | | X | 2,3,4,5,6,7,10,12 | 2,3,4,5,6,7,10,11 | 2,3,4,5,6,7,10,11,13 | | X | 2,3,4,5,6,7,10,12 | 2,3,4,5,6,7,10,11 | 2,3,4,5,6,7,10,11,13 | |  |  |  |  | |  |  |  |  | |  |  |  |  | |