Software Re-Engineering

Lecture: 14

Dr. Syed Muazzam Ali Shah

Department of Software Engineering

National University of Computer & Emerging Sciences

muazzam.ali@nu.edu.pk

Sequence [Todays Agenda]

Content of Lecture

Reverse Engineering – Techniques

- Lexical Analysis
- Syntactic Analysis
- Control Flow Analysis
- **Data Flow Analysis**
- Program Slicing
- Visualization
- Program metrics

Reverse Engineering – Techniques

Data Flow Analysis

- Data flow analysis (DFA) concerns how values of defined variables flow through and are used in a program.
  - i.e., the analysis that determines the information regarding the definition and use of data/variables in a program.
- With the help of this analysis, optimization can be done.
  - In general, it is a process in which values are computed using data flow analysis.
- The data flow property represents information that can be used for optimization.

Reverse Engineering – Techniques

Data Flow Analysis

- Data flow analysis is a technique used to analyze how data flows through a program.
- It involves tracking the values of variables and expressions as they are computed and used throughout the program.
  - with the goal of identifying opportunities for optimization and identifying potential errors

Reverse Engineering – Techniques

Data Flow Analysis

- DFA can determine data flow anomalies.
  - One example of data flow anomaly is that an undefined variable is referenced.
  - Another example of data flow anomaly is that a variable is successively defined without being referenced in between.
  - Data flow analysis enables the identification of code that can never execute, variables that might not be defined before they are used, and statements that might have to be altered when a bug is fixed.

Reverse Engineering – Techniques

Data Flow Analysis

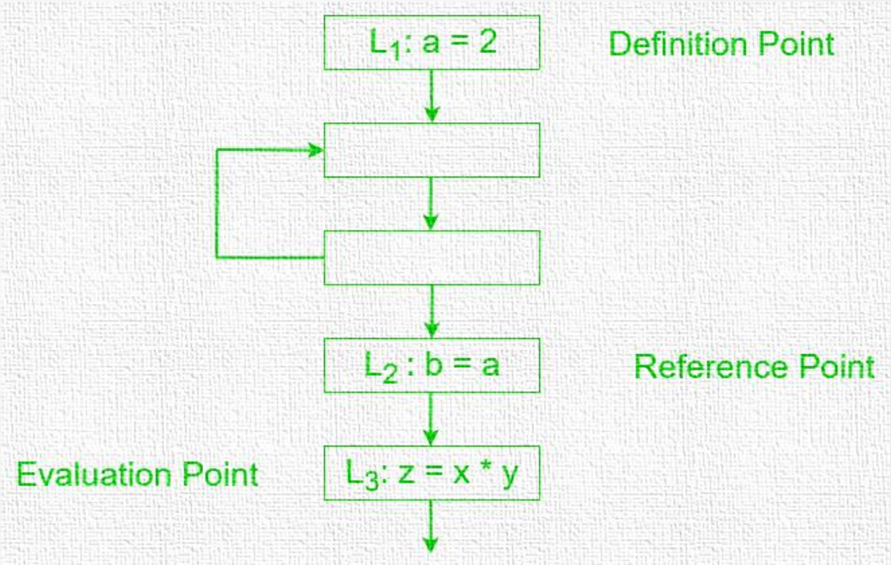
- The basic idea behind data flow analysis is to model the program as a graph,
  - Where the nodes represent program statements and the edges represent data flow dependencies between the statements.
- The data flow information is then propagated through the graph.
  - Using a set of rules and equations to compute the values of variables and expressions at each point in the program.

Reverse Engineering – Techniques

Data Flow Analysis

Basic Terminologies:

- **Definition Point:** A point in a program containing some definition.
- **Reference Point:** A point in a program containing a reference to a data item.
- **Evaluation Point:** A point in a program containing evaluation of expression.



Reverse Engineering – Techniques

Data Flow Analysis

❖ Data flow Properties

⌗Available Expression:

- An expression (a+b) is said to be available at a program point ‘x’, if none of its operands gets modified before their use.

⌗Advantage:

- It is used to eliminate common sub expressions.

```
➢ B1: L1 = 4 * i;
➢ B2: y = x + L1;
➢ B3: P = L1 * 3;
```

- The expression **L1 = 4 * i** is an available expression for **B2,** and **B3**, because **B2 and B3** are directly using expression **(L1)** at **B1**, and NONE of the operand of L1 is modified before it is used in **B2** and **B3**.

Reverse Engineering – Techniques

Data Flow Analysis

❖Data flow Properties

⌗Available Expression:

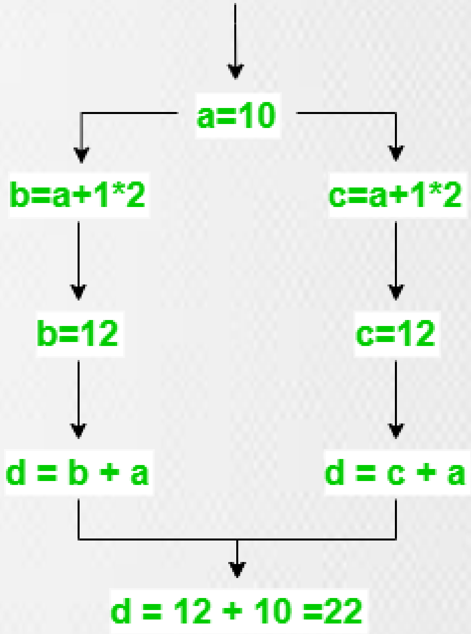
An expression is said to be available at a program point x if along paths its reaching to x. The Expression is available at its evaluation point.

After elimination

- a = 10;
- b = a + 1 * 2;
- c = a + 1 * 2;
- //'c' has common expression as 'b'
- d = c + a;

After elimination

- a = 10;
- b = a + 1 * 2;
- d = b + a;



Reverse Engineering – Techniques

Data Flow Analysis

❖Data flow Properties

⌗Reaching Definition:

- A Definition D is reaching to a point x if D is not killed or redefined before that point.

⌗Advantage:

- It is used in constant and variable propagation.

➢(It means if reaching definition of all variables have same assignment which assigns a same constant to the variable, then the variable has a constant value and can be substituted with the constant).

- ➢ `B1: D1: x = 4;`
- ➢ `B2: D2: x = x + 2;`
- ➢ `B3: D3: y = x + 2;`

- **D1** is reaching definition for **B2**, however not for **B3** Since it is killed by **B2**.

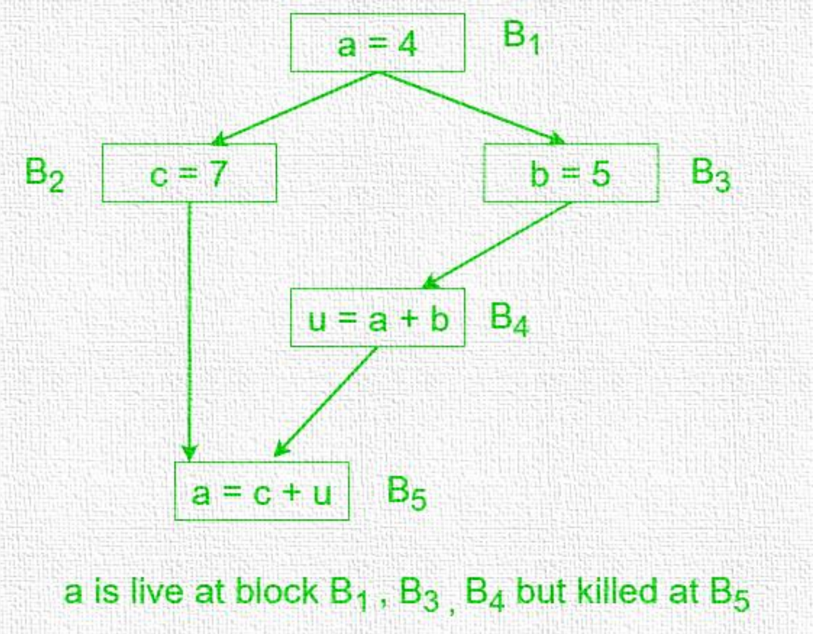
Reverse Engineering – Techniques

Data Flow Analysis

- Data flow Properties

Live Variable:

- A variable is said to be live at some point p if from p to end the variable is used before it is redefined else it becomes dead/killed.
- It is used in dead code elimination.



a is live at block B_1 , B_3 , B_4 but killed at B_5

Reverse Engineering – Techniques

Data Flow Analysis

- Data flow Properties

Busy Expression :

- An expression is busy along a path if its evaluation exists along that path and none of its operand definition exists before its evaluation along the path.

Reverse Engineering – Techniques

Data Flow Analysis

- Features

- **Identifying dependencies:**
  - Data flow analysis can identify dependencies between different parts of a program, such as variables that are read or modified by multiple statements.
- **Detecting dead code :**
  - By tracking how variables are used, data flow analysis can detect code that is never executed, such as statements that assign values to variables that are never used.

Reverse Engineering – Techniques

Data Flow Analysis

- Features

- **Optimizing Code:**
  - Data flow analysis can be used to optimize code by identifying opportunities for common subexpression elimination, constant folding, and other optimization techniques.
- **Detecting Errors:**
  - Data flow analysis can detect errors in a program, such as uninitialized variables, by tracking how variables are used throughout the program.

Reverse Engineering – Techniques

Data Flow Analysis

- Data flow analysis can have a number of advantages

- **Improved code quality:**
  - By identifying opportunities for optimization and eliminating potential errors, data flow analysis can help improve the quality and efficiency of the compiled code.

- **Better error detection:**
  - By tracking the flow of data through the program, data flow analysis can help identify potential errors and bugs that might otherwise go unnoticed.

Reverse Engineering – Techniques

Data Flow Analysis

- Data flow analysis can have a number of advantages

Increased understanding of program behavior:

- By modeling the program as a graph and tracking the flow of data:
    - Data flow analysis can help programmers better understand how the program works and how it can be improved.

Thank You!