Enhanced Representation Of Data Flow Anomaly Detection For Teaching Evaluation

T.Mamatha  
Asst.Prof. in Dept. of CSE  
SreeNidhi Institute of Science & Technology  
Yammampet, Ghatkesar,Hyderabad, TS,INDIA-501301  
E-mail: mamathat7@gmail.com

A.BalaRam  
Assoc.Prof. in Dept of CSE  
CMR Institute of Technology  
Medchal, Hyderabad,Ts,India -501 401  
E-Mail:balarambalaram@gmail.com

Dr. D.S.R. Murthy  
Prof. & Head of CSE  
Geethanjali College of Engg & Tech  
Hyderabad-501301-TS-INDIA  
Email :dsmurthy.1406@gmail.com

N. Archana  
Asst.Prof. in Dept of CSE  
Sreenidhi Institute of Science & Technology  
Yamnampe, Ghatkesar, Hyderabad,TS,INDIA-1301  
Email: archana.nagelli@gmail.com

ABSTRACT
In this paper we propose three letter sequence of action for dataflow anomaly detection and representing data flow model for each data object. When we represent the CFG for data object, it is not giving the information that the object is used for which purpose(whether for calculation, predicate or redefined) so we extend the representation for more understanding and also given the DFD for all possible situation for normal flow and anomaly flow.

KEYWORDS: Data Flow Testing, Data flow Anomaly Detection, Data flow model, control flow graph

1. INTRODUCTION
Data-flow testing is the name given to a family of test strategies based on selecting paths through the program’s control flow in order to explore sequences of events related to the status of data objects. For example, pick enough paths to assure that every data item has been initialized prior to its use, or that all objects have been used for something, if not then it is anomalous.

• Anomaly, it is an abnormal way of doing something.

Example:
\[ x = f_1(y); \]
\[ x = f_2(z); \]
The second definition of x overrides the first
• Three types of abnormal situations with using variable.
  – Type 1: Defined and then defined again
very easy to understand and with a little practice, reasonably quick and straightforward to develop.

Program actions are represented by d, k and u as below:

Program Actions  (d, k, u):

Defined (created) - explicitly or implicitly (d)
Killed (released ) - directly or indirectly  (k)
Used - directly(or)indirectly (u)

In a calculation - (c)
In a predicate - (p)

2. THREE LETTER SEQUENCE OF ACTIONS

When your using data object for some purpose and representing using control flow graph or Data flow graph, It is not showing that whether the data object is used for calculation purpose, predicate purpose or for redefining. Here we extend the representation by using the notation as follows.

- D is for Defined, initialized or created (explicitly or implicitly)
  - Redefined - Dr
- K is for killed or released (directly or indirectly)
- U for Usage
  - In calculation - Uc
  - In Predicate - Up

Uc : Represents that object or variable is used for calculation purpose
Up : Represents that object or variable is used for Predicate purpose.
Dr : Represents that the object or variable is redefined

The task or action to be performed on the object or variable is denoted by d, k and u (d for define, u for usage and k for killing). It follow sequence of task like first it should be defined then used for some purpose, lastly it can kill.

- indicates the nothing is happened on the current time

Example: d-k indicates that object is defined and killed without using the object for any purpose. It giving more understanding of what happened to data object or variable.

2.1 Proposed  Three letter sequence of Actions(d,u,k) :

duk - normal
duu - normal
dku - Anomaly
dkd - Probably a bug
ddu - normal
dud - normal
udk - Anomaly
ukd - Anomaly
udd - Anomaly
ukk - Anomaly
uuk - Anomaly
kdu - harmless, but suspicious
kdd - normal
kud - Anomaly
kuu - Anomaly
kku - Anomaly
ukd - probably a Anomaly
udd - Anomaly
ukk - Anomaly
uuk - Anomaly
kdu - harmless but suspicious

Leading dash and Trailing dash means, nothing happened form entry to current and trailing dash to mean nothing happened after that point.

d- - Harmless, Suspicious, defined but not used for any purpose after that point

duk - Normal, Defined and used for some Purpose.
d-k - Probably a Bug, Defined but Without using killed the object.

3. REPRESENTATION OF DATA OBJECT USING CONTROL FLOW GRAPH

Constructing a simplified control flow graph from the given structured code is:

- Create a simplified flow graph for code given.
- For More understanding, Constructed flow graph for each data object and mentioned whether object is used for predicate purpose or Calculation Purpose.

Example:

\[
\begin{align*}
Z &= b + \ldots \ldots \\
&= \frac{an - 1}{a - 1}
\end{align*}
\]
START
INPUT a, b, n
Z := 0
IF a = 1 THEN Z := 1
GOTO DONE1
r := 1 c := 1
POWER:
c := c * a
r := r + 1
IF r <= n THEN GO TO POWER
Z := (c - 1) / (a - 1)
DONE1:
Z := b + Z
END

Fig 1: simplified flow graph for above program

3.1 Control Flow Graph for Each Data Object:

Fig 2: control flow graph for the object Z:

Fig 3: Control flow graph for the object c:

Fig 4: Control flow graph for the object r:

Fig 5: control flow graph for the object b:

Fig 6: control flow graph for the object n:
5. CONCLUSION

The main objective of this enhancement is for better understanding of the data object usage. When the object is defined(d), but not yet used(U) or killed(k) It becomes anomalous, this kind of representation with two letter sequence have not proper understanding. So we extend the representation for more understanding. We propose flow graph for each data object to show how the object is used after it is defined. And finally we depict State graph, shows the anomalous states and some Normal states.

6. REFERENCES


Authors


Mr. A. Balaram working as Associate Professor in the Dept. Of Computer Science and Engineering, CMR Institute of Technology, Hyderabad, India. He has more than 10 years of teaching experience. His research interests are Software Engineering, Image Processing, Net work Security and Cryptography.

Dr. D. S. R. Murthy is currently working as a Professor & Head of the Dept of CSE in Geethanjali college of Engineering and Technology since June 2014. Earlier he worked as a professor of Information Technology in Sreenidhi Institute of science and Technology from OCT 2004 to JUNE 2011. Prior to that he was in NIT Warangal JNTUCE, Anantapur and ICFAI in different faculty positions of Computer Science. He is a Fellow of IETE, Senior Life Engineer (El(I) & IETE). He published a text book on C Programming & Data Structures. His research interests are Image Processing, Image Cryptography and Data Mining.

Mrs. N. Archana, Working as an Assistant Professor in Dept of C.S.E of Sreenidhi Institute of Science & Technology. M.Tech in Software Engineering from JNTU, Hyderabad. B.Tech in C.S.E from JNTUH. She has more than 8 yrs of experience. Her Area of Interest is Wireless Sensor Network & Information Security.