National Aeronautics and Space Administration

Differential Symbolic Execution (DSE)

Suzette Person[1], **Matthew B. Dwyer**[2], **Sebastian Elbaum**[2], **Corina Păsăreanu**[3]

[1]NASA Langley Research Center, [2]University of Nebraska-Lincoln, [3]NASA Ames Research Center

SOFTWARE EVOLVES

C130J

- Three year development effort:*
  - **6895** revisions
  - **870** builds

- Fix defects
- Extend functionality
- Refactor to improve maintainability
- Improve performance
- Add new features

Apache — 10K revisions per product (average)

VALIDATING EVOLVING SOFTWARE

- Safety-critical applications require rigorous re-validation to ensure safety of human lives and to avoid financial loss
- System size and complexity make re-validation costly and time consuming

2.25M test cases*

28 months to certify operational increment**

FOCUSED RE-VALIDATION

Reduce re-validation cost and effort by focusing on program *differences*

```
public int logicalValue(int t){
  if (t[currentTime - t >= 100]){
    return old;
  }else{
    int val = 0;
    for (int i=0; i<data.length; i++){
      val = val + data[i];
    }
    old = val;
    return val;
  }
}
```

```
  final int THRESHOLD = 100;
public int logicalValue(int t){
  int elapsed = currentTime - t;
  int val = 0;
  if (elapsed < THRESHOLD){
    val = old;
  }else{
    for (int i=0; i<data.length; i++){
      val = val + data[i];
    }
    old = val;
  }
  return val;
}
```

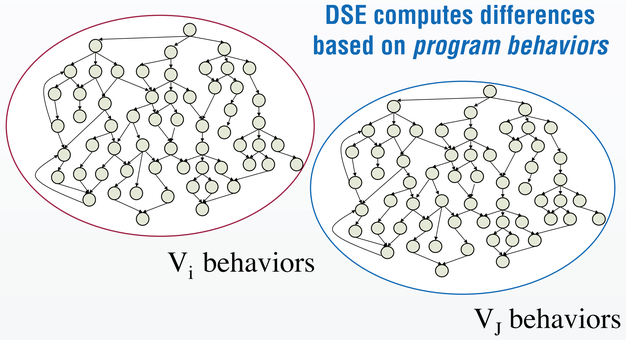
Source-based differencing techniques identify *location* of program differences

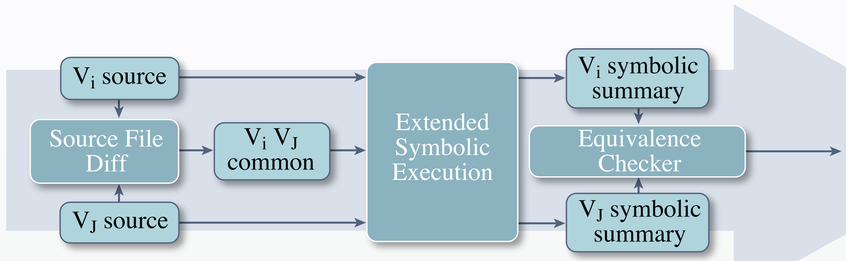
✔ Widely available	✘ Imprecise
✔ Inexpensive	✘ May miss changed behaviors

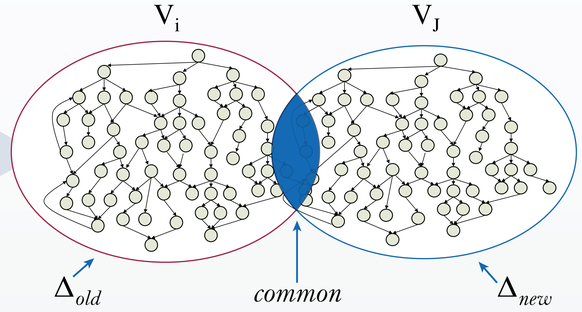
Sensitive to formatting and minor syntactic changes

DIFFERENTIAL SYMBOLIC EXECUTION

DSE computes differences based on *program behaviors*



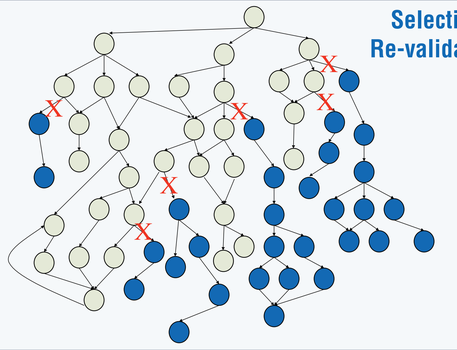




- ✔ **More precise than source based differencing techniques**
- ✔ **Leverages program commonalities**
- ✔ **Useful for a wide range of software evolution tasks**

❖ Selective Re-validation	❖ Test Suite Evolution
❖ Change Characterization	❖ Impact Analysis
❖ Test Case Selection/Prioritization	❖ …
❖ Refactoring Assurance	

Selective Re-validation

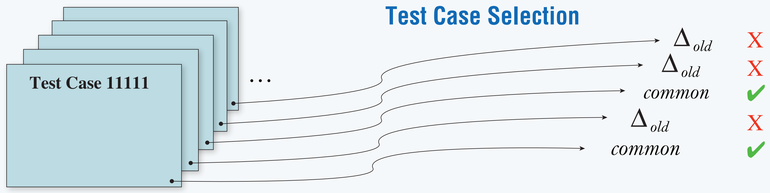


CLIENT ANALYSES

Change Characterization

On input	match() Version 3	match() Version 4
x == null && y == null	throws *java.lang.NullPointerException*	returns TRUE
x == null && y !=null	throws *java.lang.NullPointerException*	returns FALSE
x != null && y == null	throws *java.lang.NullPointerException*	returns FALSE

Test Case Selection



This work was supported in part by the National Aeronautics and Space Administration under grant number NNX08AV20A, the National Science Foundation under award CNS-0454203, and by a Google Anita Borg Memorial Scholarship.

* Richard Conn, Stephen Traub, and Steven J. Chung. Avionics modernization and the C-130J software factory. CrossTalk: The Journal of Defense Software Engineering, September 2001.

** Marvin V. Zelkowitz and Ioana Rus. The role of independent verification and validation in maintaining a safety critical evolutionary software in a complex environment: The NASA space shuttle program. In Proceedings of the International Conference on Software Maintenance, page 118-126, 2001.
Marvin V. Zelkowitz and Ioana Rus. Understanding IV & V in a safety critical and complex evolutionary environment: The NASA space shuttle program. In Proceedings of the 23rd International Conference on Software Engineering, pages 349-357, 2001.