

Incremental Test Case Generation for UML-RT Models Using Symbolic Execution



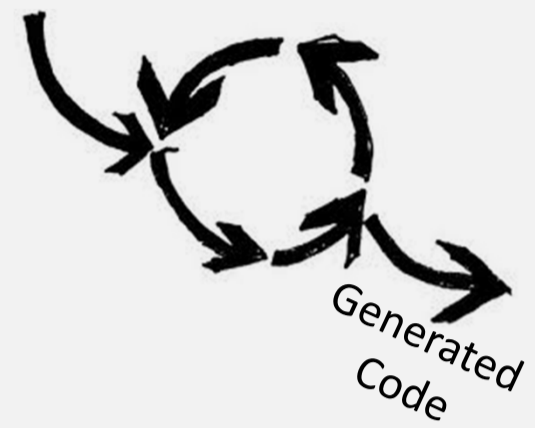
Eric James Rapos, Juergen Dingel
 {eric,dingel}@cs.queensu.ca
 Modeling & Analysis in Software Engineering Group
 School of Computing
 Queen's University, Kingston, Ontario, Canada



Ontario Centres of Excellence
 Where Next Happens

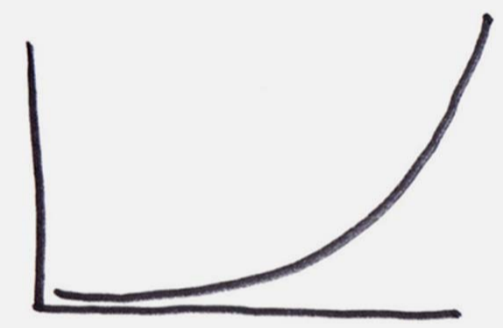


Motivation and Goals



The Iterative Nature of Model-Driven Development (MDD)

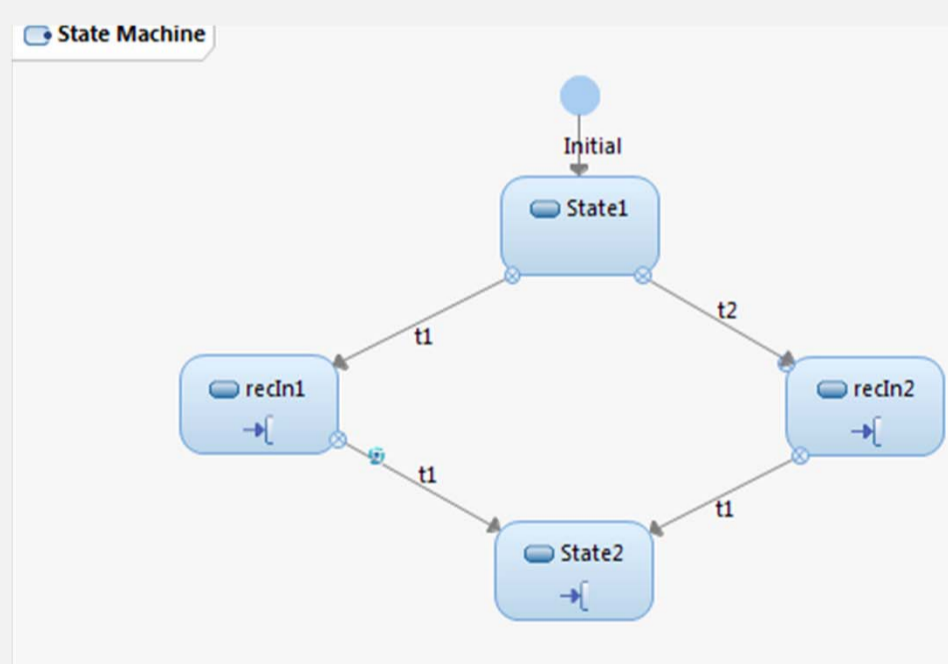
Understanding and Classifying the Effects of Model Evolution on Execution and Testing



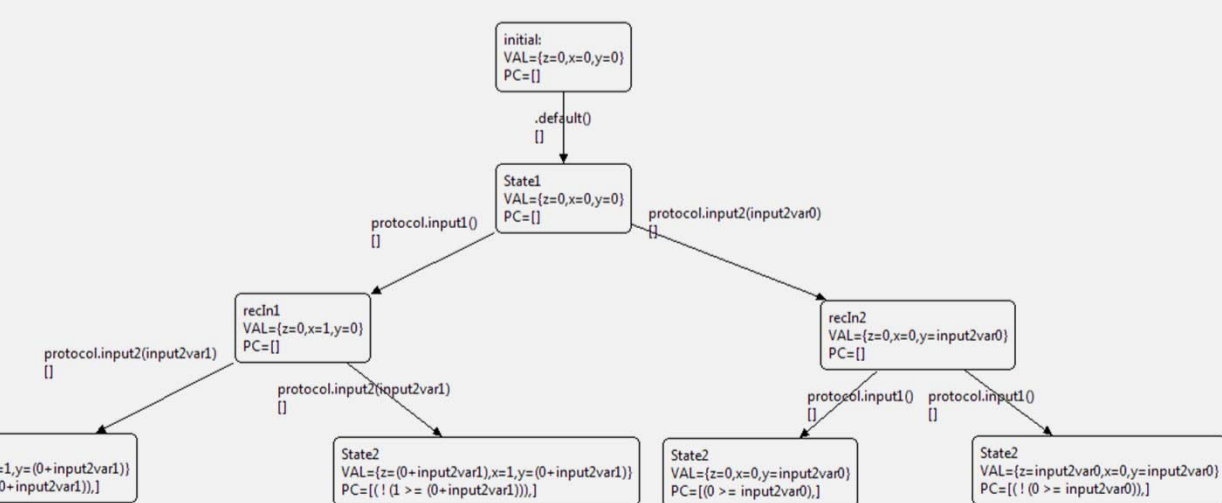
Improving Efficiency of Test Case Generation Tools by Reducing Redundancy

Background

Real-Time Software Modeling (UML-RT)

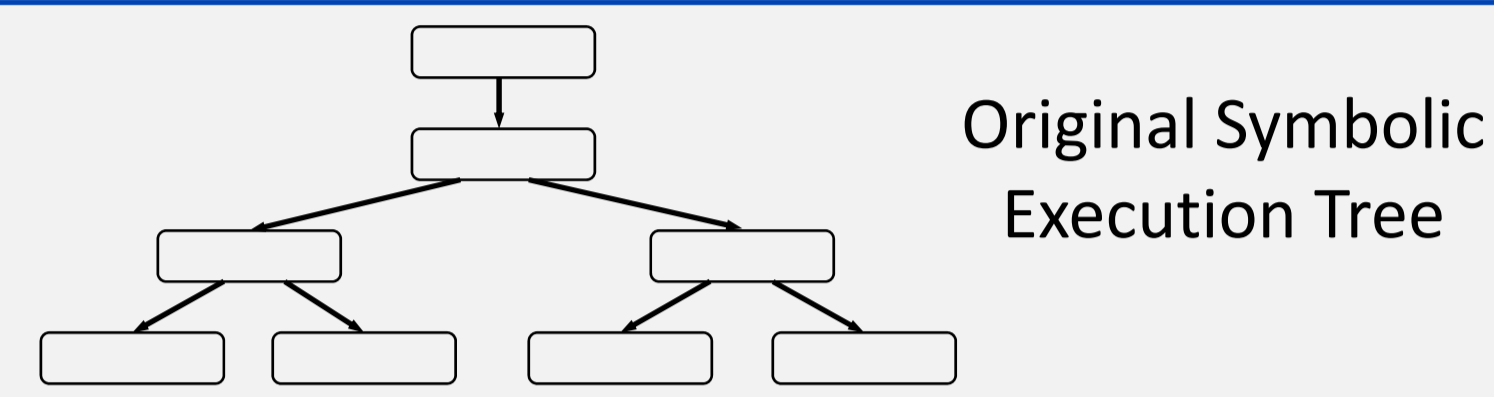


Symbolic Execution of UML-RT State Machines

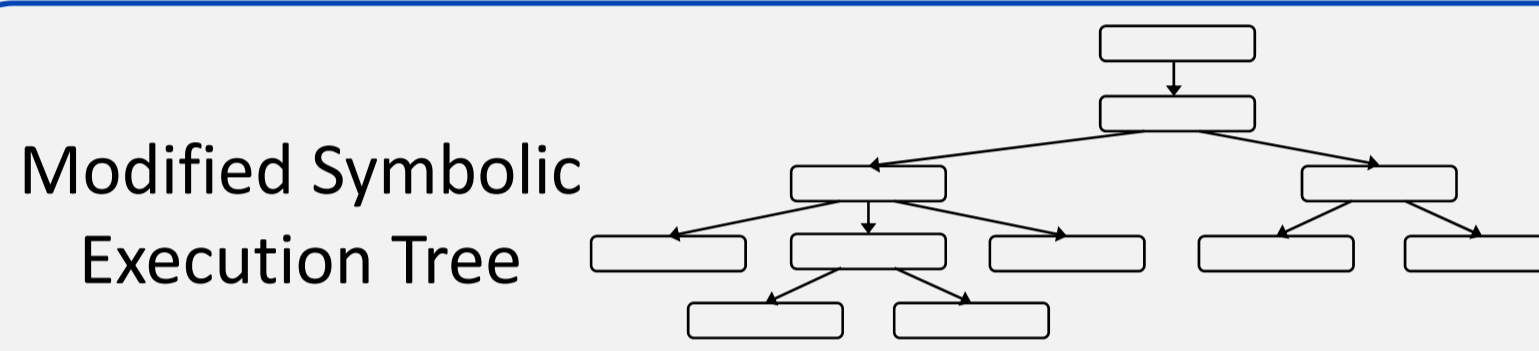


Process

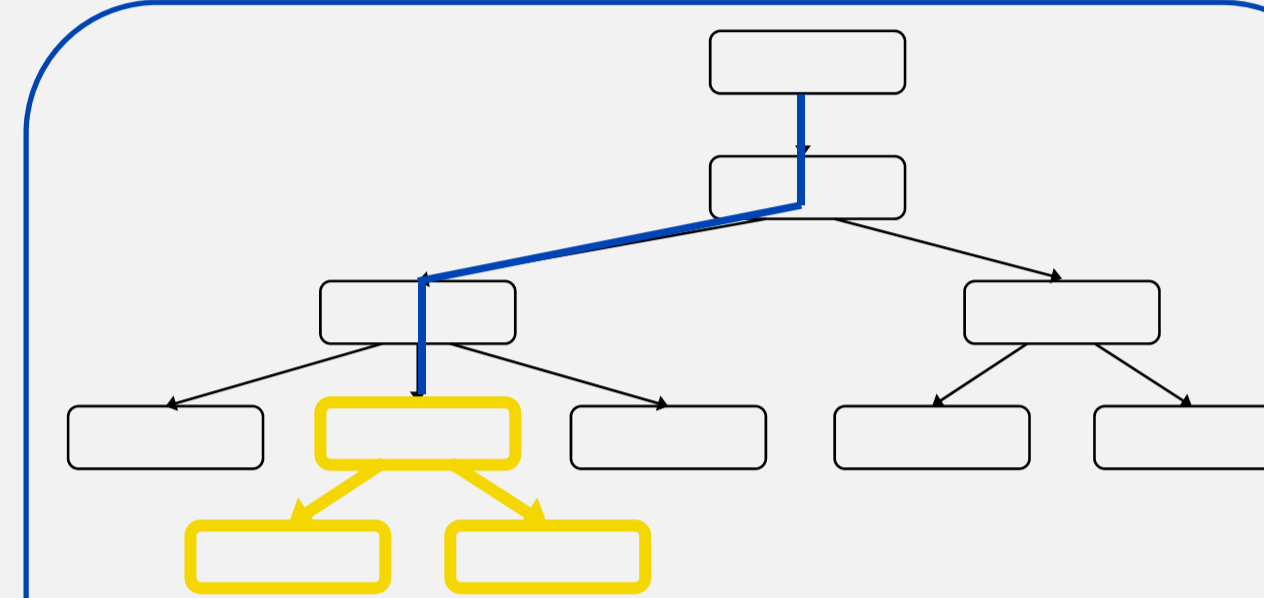
1 Differencing Symbolic Execution Trees (SETs)



Original Symbolic Execution Tree

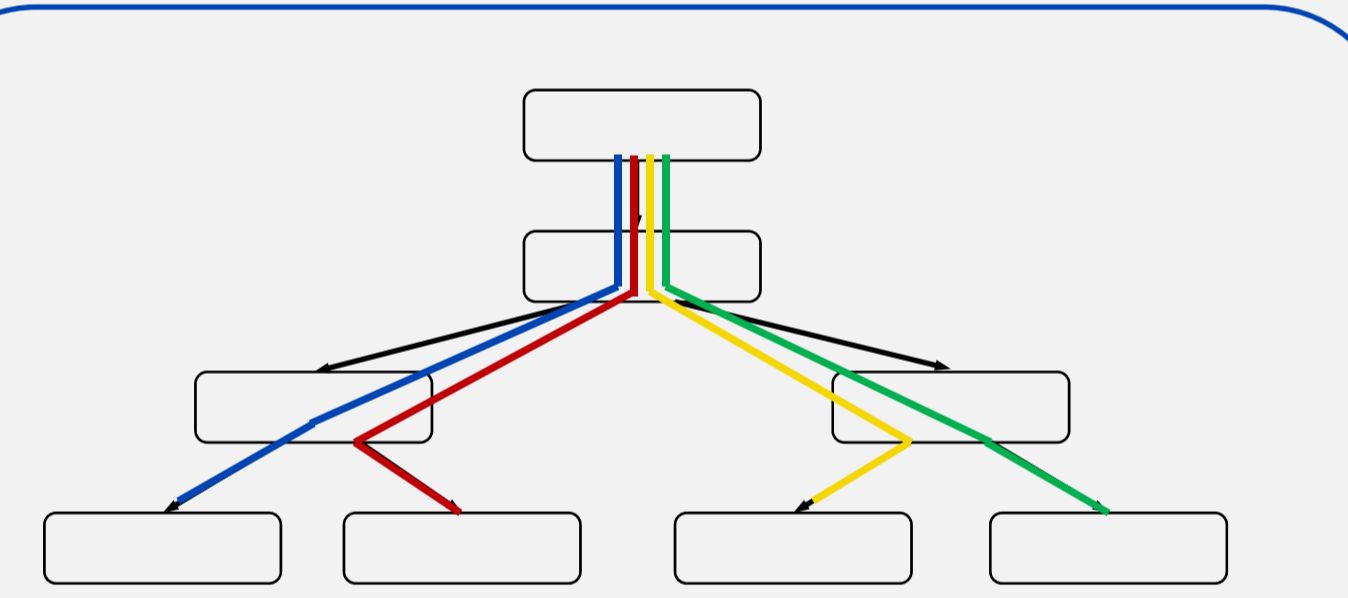


Modified Symbolic Execution Tree

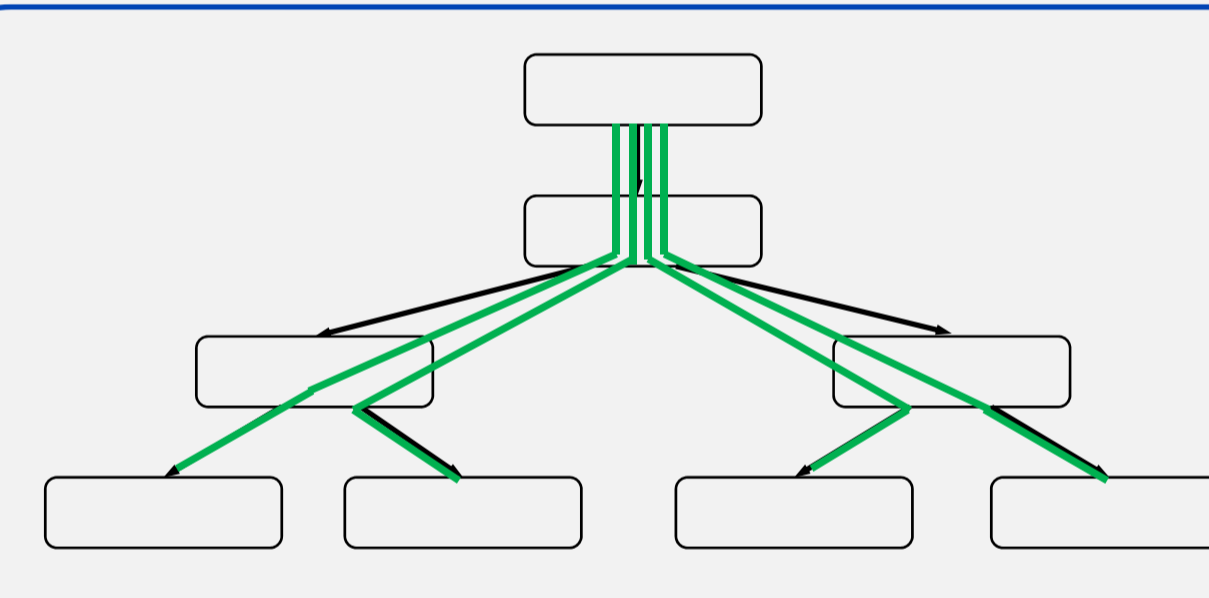


Determine Paths (in blue) to Tree Differences (in yellow)

2 Initial Test Case Generation & Examination

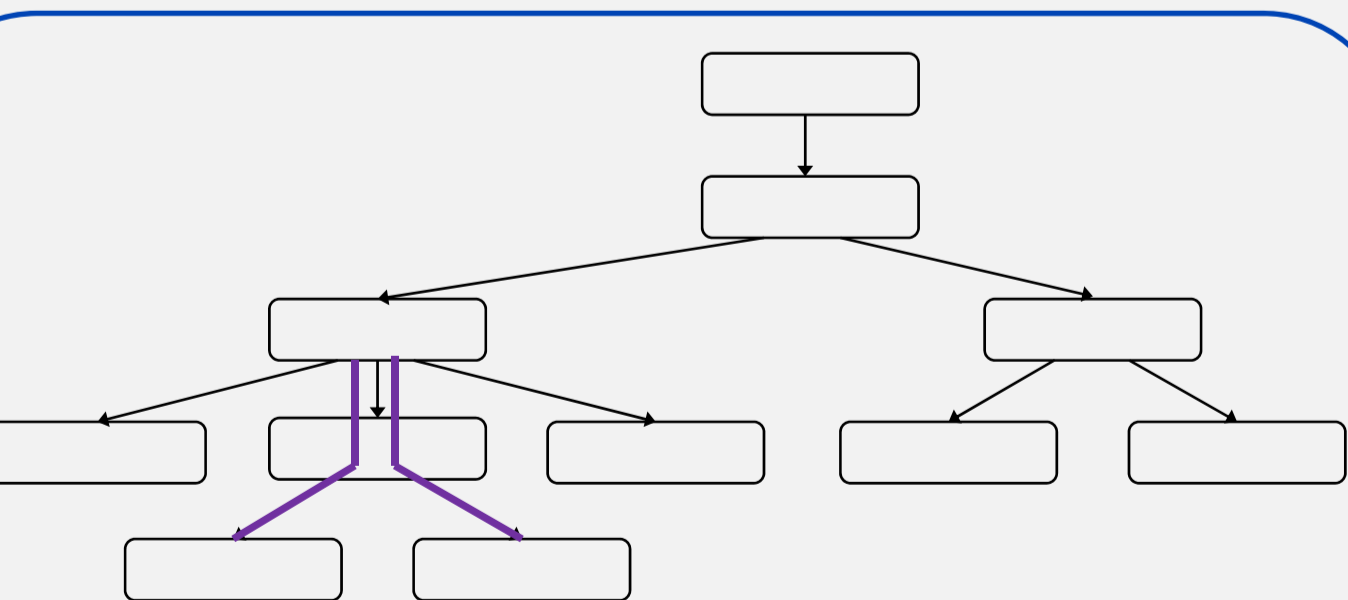


Path Coverage Based Test Case Generation

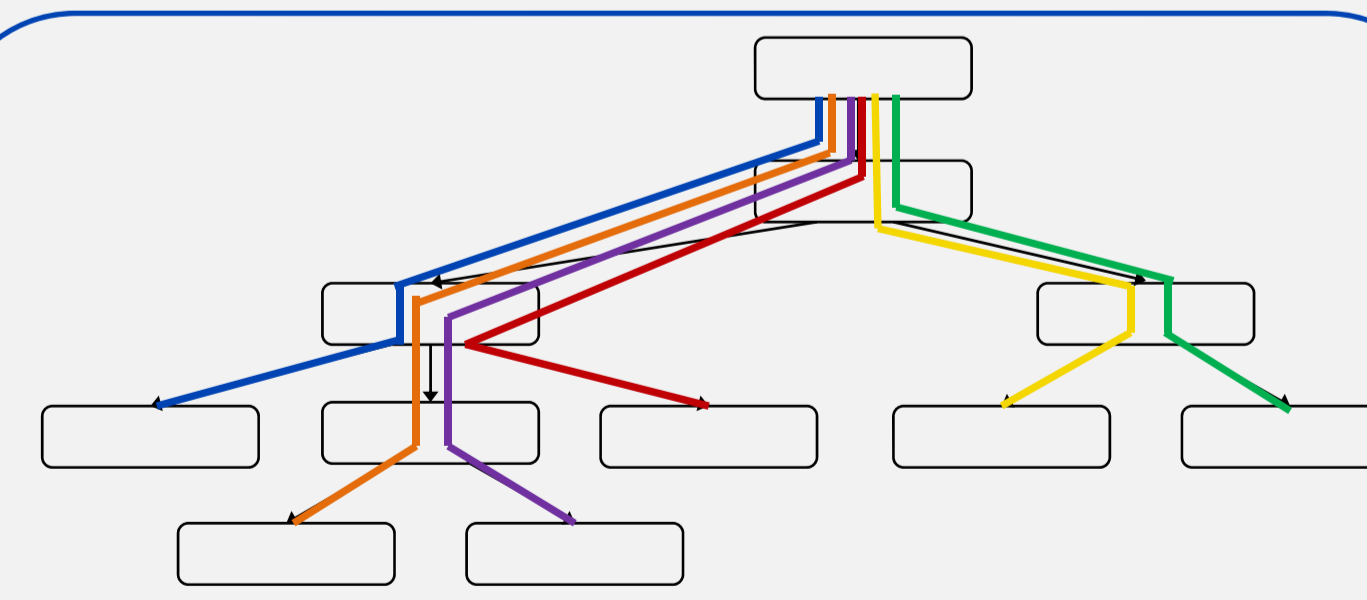


Keep Test Cases That Are Not Affected by Differences (in green) – Remove Others

3 Incremental Generation of New Test Cases

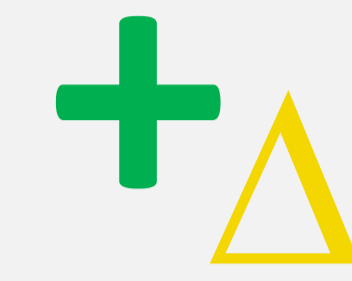


Generate New Paths From Highest Difference (in purple)



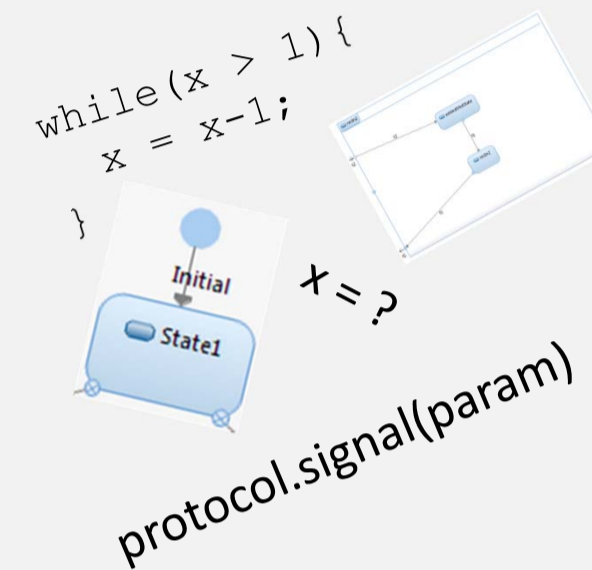
Add Prefix (blue from Step 1) to New Paths to Generate Full Tests & Add to Final Test Suite

Evolution Steps



Three Main Areas of Focus: Additions, Modifications and Deletions

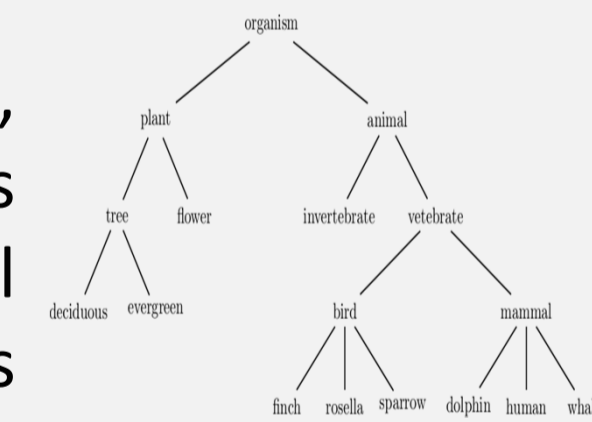
Looking at States, Transitions, Parameters, Attributes, Action Code, and Hierarchy



Having examples from each area for each artifact ensures a coverage of all logical evolution steps

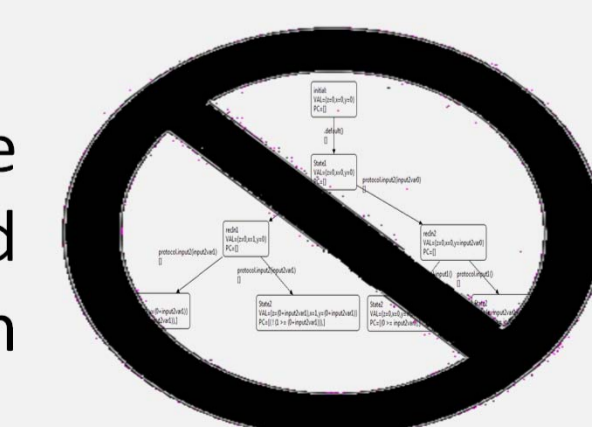
Planned Work

Using Empirical Results, Create a Set of Classifications of the effects of Model Evolution on Test Cases



Develop an Integrated Plugin for Rational Software Architect – Real-Time Edition (RSA-RTE)

Use the Classifications of the Effects to Minimize the Need for Symbolic Execution



Resources

- [ZD12] K. Zurowska and J. Dingel. "Symbolic Execution of UML-RT State Machines". 27th ACM Symposium on Applied Computing, Track on Software Verification and Testing (SAC-SVT'12). Riva del Garda, Italy, March 25-29, 2012.
- [ZD11] K. Zurowska and J. Dingel. "SAUML - a Tool for Symbolic Analysis of UML-RT Models". Tool Demonstration Paper. 26th IEEE/ACM International Conference On Automated Software Engineering (ASE'11). Lawrence, Kansas, USA, Nov 6-10, 2011.
- [UKB10] E. Uzuncaova, S. Khurshid, D. S. Batory. "Incremental Test Generation for Software Product Lines". IEEE Transactions on Software Engineering; 36(3): 309-322 (2010)
- [VPK04] W. Visser, C. S. Păsăreanu, S. Khurshid. "Test input generation with java PathFinder". 2004 ACM SIGSOFT international symposium on Software testing and analysis (ISSTA '04)
- [FWPG07] B. Fluri, M. Wursch, M. Pinzger, H.C. Gall. "Change Distilling: Tree Differencing for Fine-Grained Source Code Change Extraction". IEEE Transactions on Software Engineering; 33(11):725-743 (2007)
- IBM Rational Software Architect Real-Time Edition (RSA-RTE) - http://www-947.ibm.com/support/entry/portal/Overview/Software/Rational/Rational_Software_Architect_RealTime_Edition
- Eclipse Modeling Framework (EMF) - <http://www.eclipse.org/modeling/emf/>
- Choco Constraint Solver - <http://choco.emn.fr/>