Discriminating Influences among Instructions in a Dynamic Slice

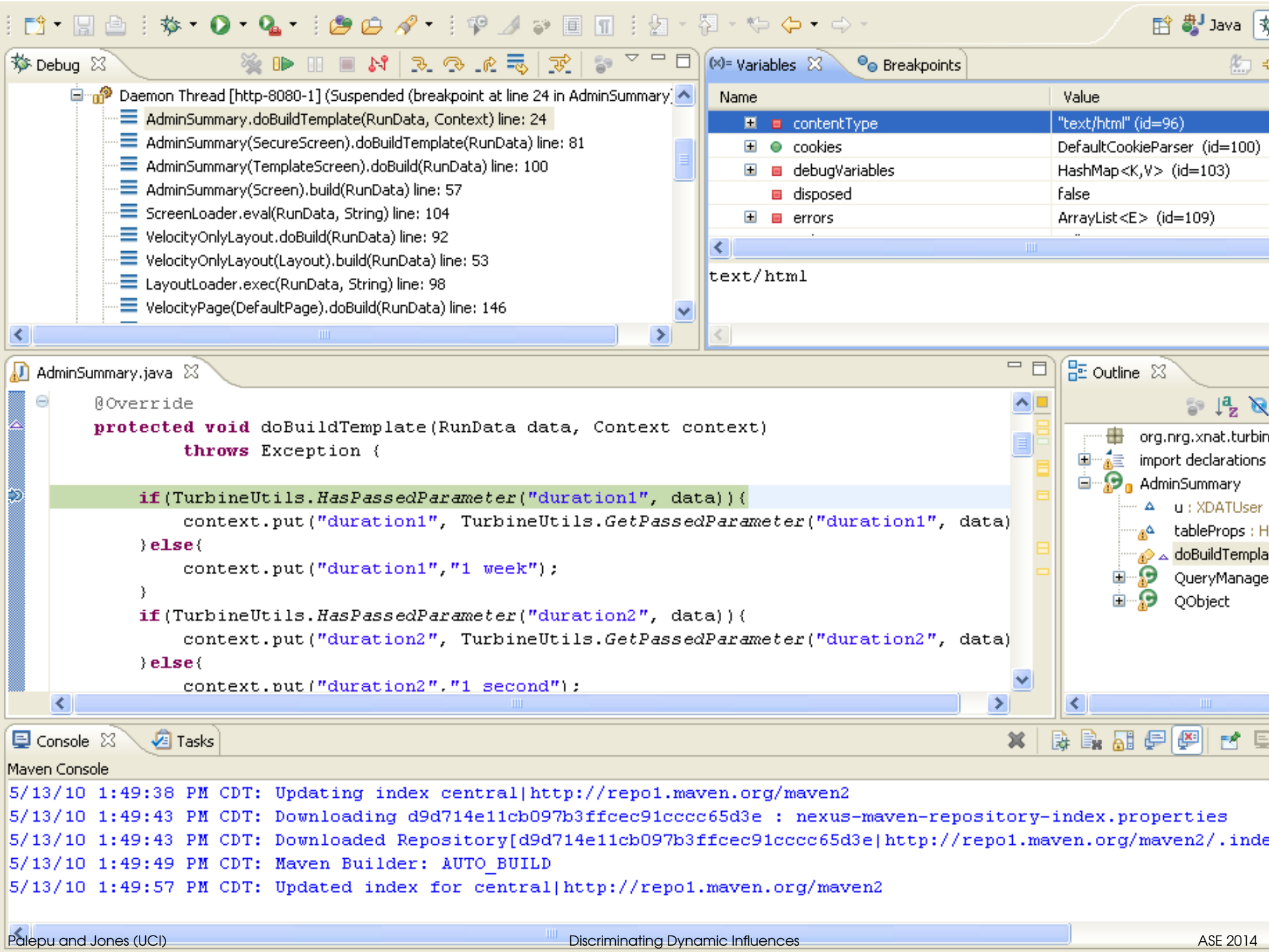
Vijay Krishna Palepu James A. Jones

University of California, Irvine, USA

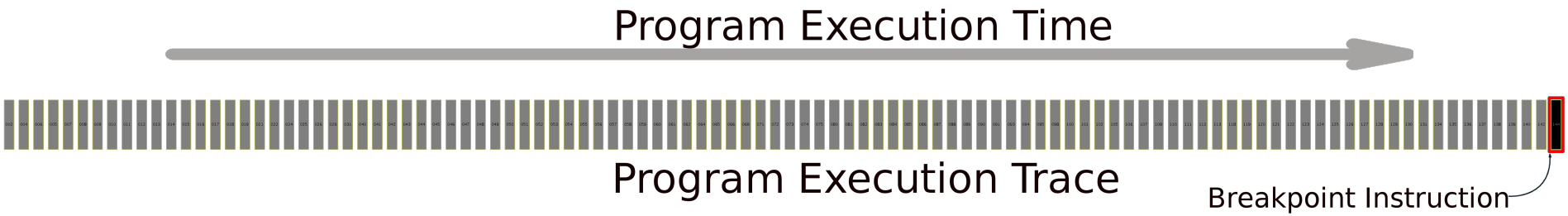
29th IEEE/ACM International Conference on Automated Software Engineering, 2014

UC Irvine

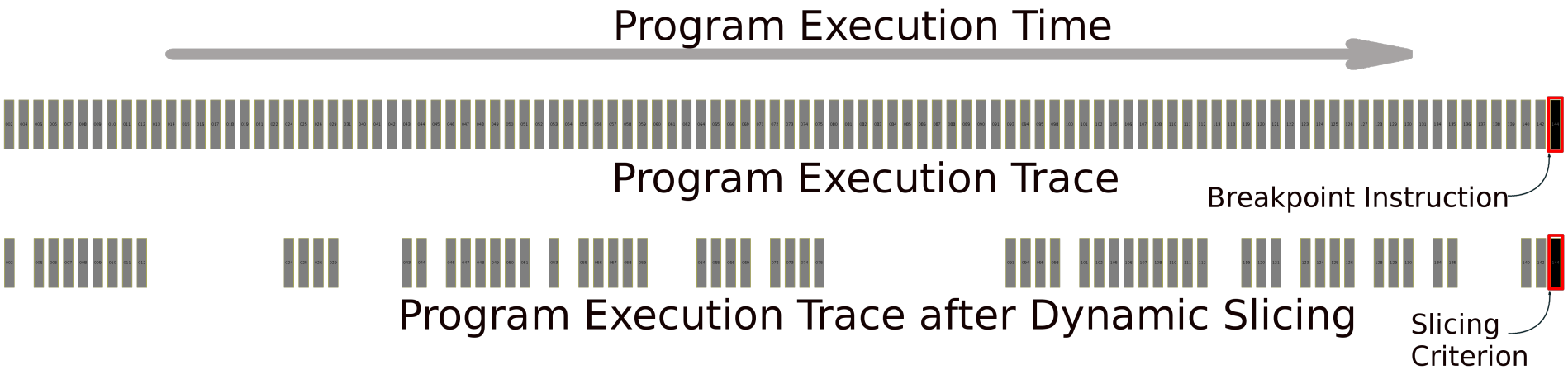
Spider Lab


Debug
Daemon Thread [http-8080-1] (Suspended (breakpoint at line 24 in AdminSummary
AdminSummary.doBuildTemplate(RunData, Context) line: 24
AdminSummary(SecureScreen).doBuildTemplate(RunData) line: 81
AdminSummary(TemplateScreen).doBuild(RunData) line: 100
AdminSummary(Screen).build(RunData) line: 57
ScreenLoader.eval(RunData, String) line: 104
VelocityOnlyLayout.doBuild(RunData) line: 92
VelocityOnlyLayout(Layout).build(RunData) line: 53
LayoutLoader.exec(RunData, String) line: 98
VelocityPage(DefaultPage).doBuild(RunData) line: 146
Variables
Breakpoints
Name
Value
contentType
"text/html" (id=96)
cookies
DefaultCookieParser (id=100)
debugVariables
HashMap<K,V> (id=103)
disposed
false
errors
ArrayList<E> (id=109)
text/html
AdminSummary.java
@Override
protected void doBuildTemplate(RunData data, Context context)
throws Exception {
if(TurbineUtils.HasPassedParameter("duration1", data)){
context.put("duration1", TurbineUtils.GetPassedParameter("duration1", data)
}else{
context.put("duration1","1 week");
}
if(TurbineUtils.HasPassedParameter("duration2", data)){
context.put("duration2", TurbineUtils.GetPassedParameter("duration2", data)
}else{
context.put("duration2","1 second");
Outline
org.nrg.xnat.turbin
import declarations
AdminSummary
u : XDATUser
tableProps : H
doBuildTempla
QueryManage
QObject
Console
Tasks
Maven Console
5/13/10 1:49:38 PM CDT: Updating index central|http://repo1.maven.org/maven2
5/13/10 1:49:43 PM CDT: Downloading d9d714e11cb097b3ffcec91cccc65d3e : nexus-maven-repository-index.properties
5/13/10 1:49:43 PM CDT: Downloaded Repository[d9d714e11cb097b3ffcec91cccc65d3e|http://repo1.maven.org/maven2/.inde
5/13/10 1:49:49 PM CDT: Maven Builder: AUTO_BUILD
5/13/10 1:49:57 PM CDT: Updated index for central|http://repo1.maven.org/maven2

Program Execution



Dynamic Slicing

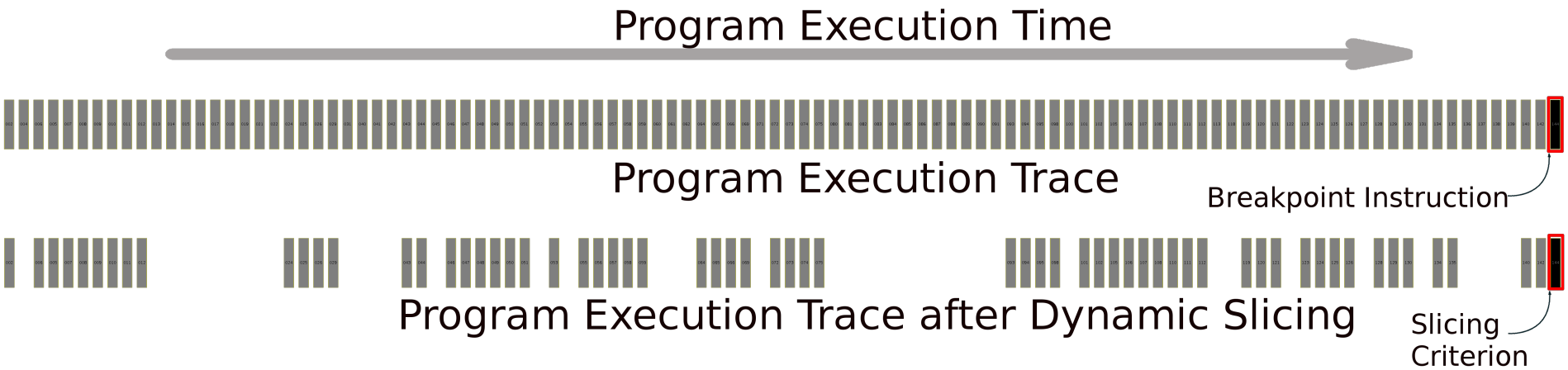


Problem.

- "... for most real programs, the dynamic slices are **too large for humans to inspect and comprehend.**" – Wang and Roychoudhury, ISSTA 2007.

- "Breakpoints allowed developers to search for paths to a statement. But **setting breakpoints was impractical when searching for many statements** ...." – LaToza and Myers, ICSE 2010

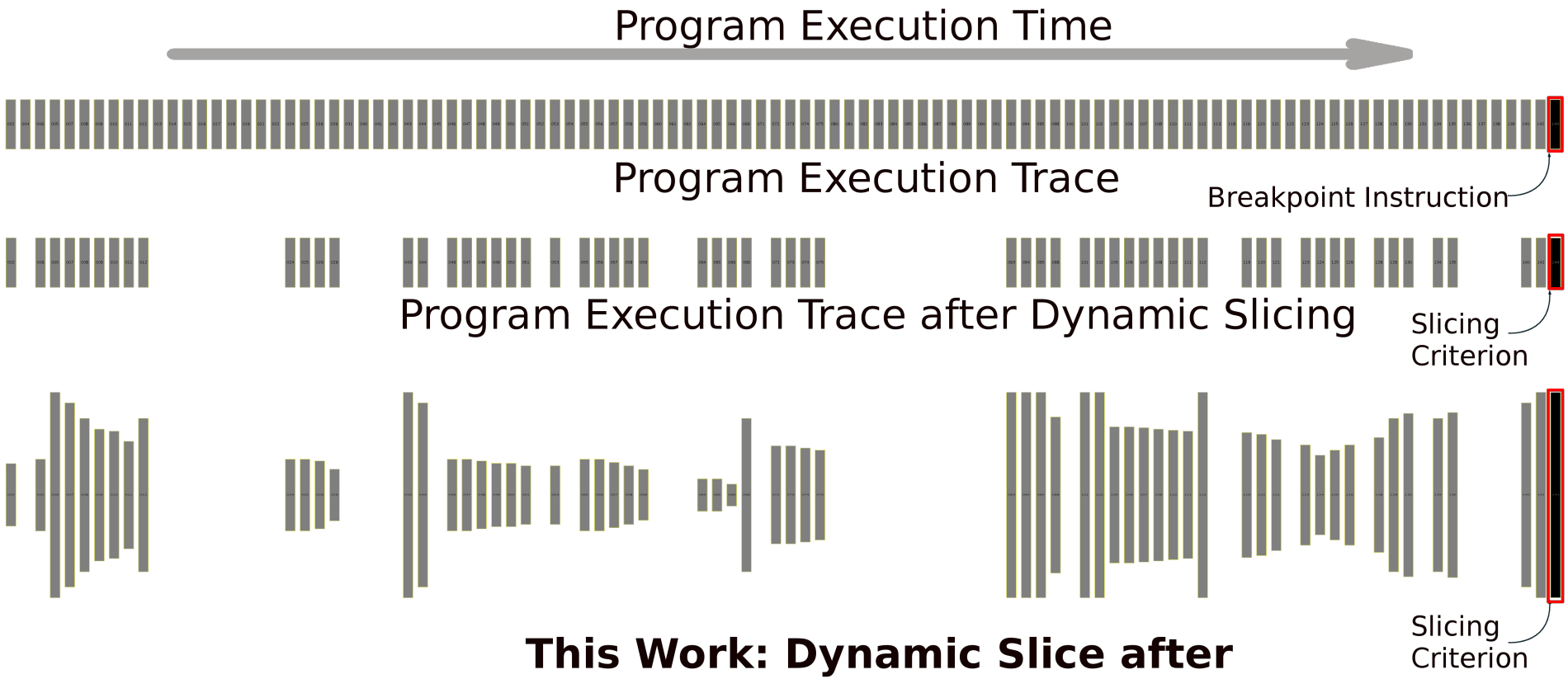
Dynamic Slicing



Problem.

- **Where, in the program's execution, should we focus our attention?**

Discrimination of Influences

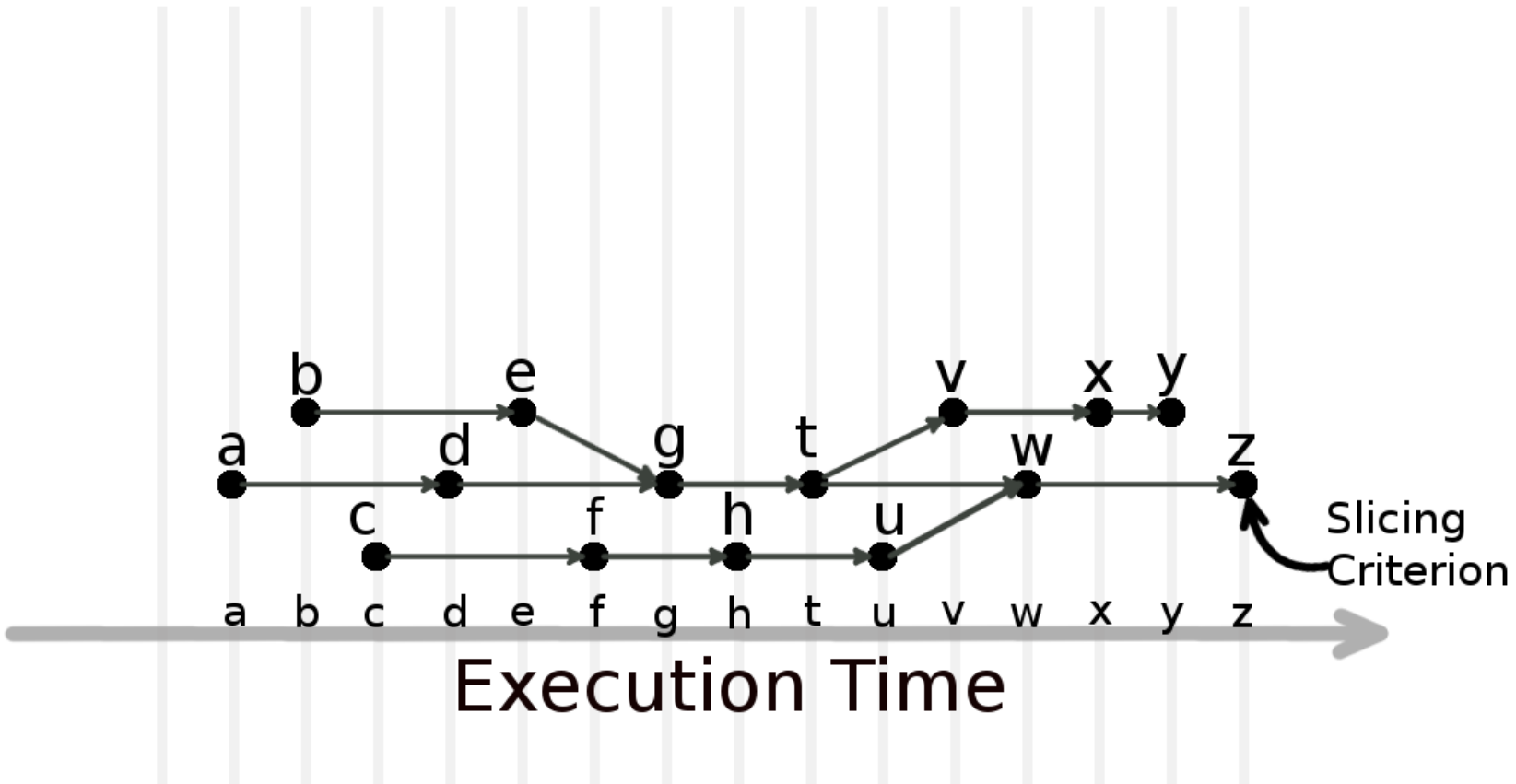


Approach.

- **Discriminate runtime instructions** using **degrees of influence** on the slicing criterion.

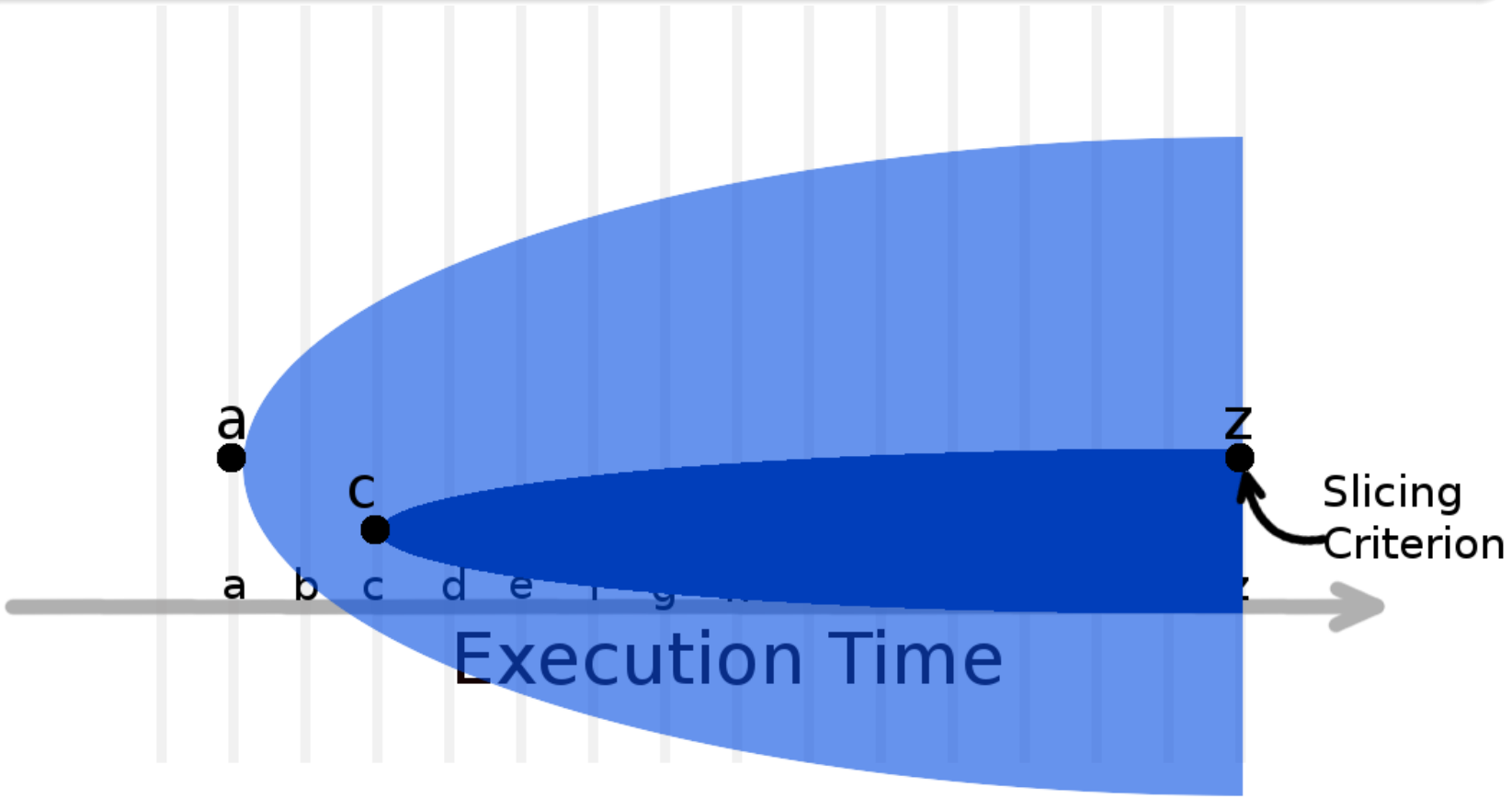
Dynamic Relevance

Measure of Degree of Influence.



Dynamic Relevance: Intuition

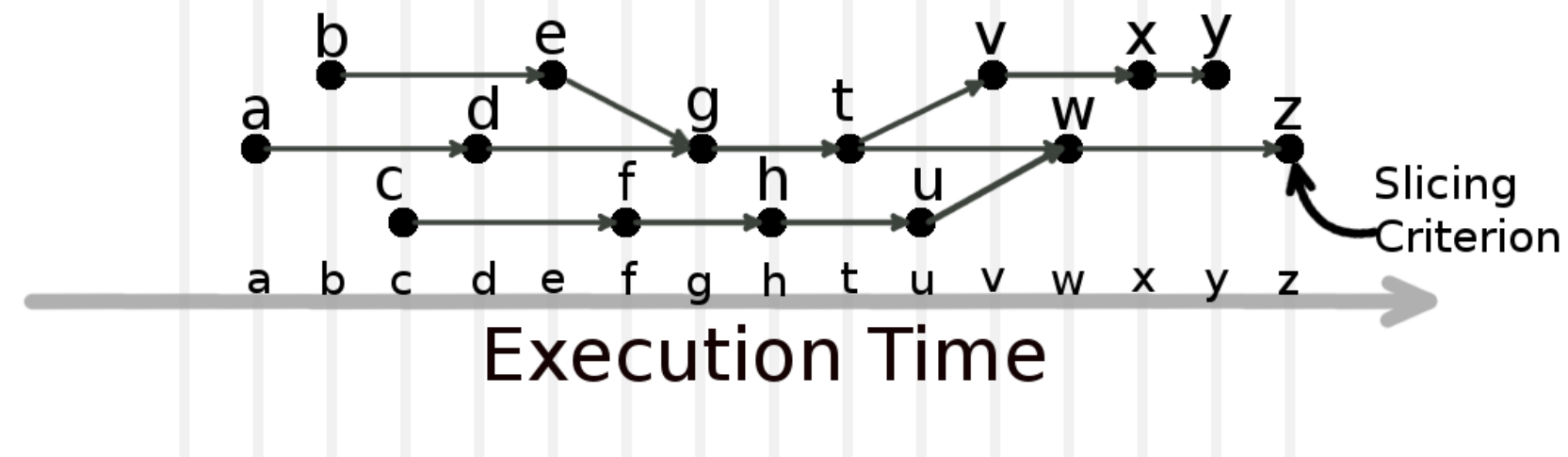
Broader the scope of influence, lesser the dynamic relevance and vice versa.



Dynamic Relevance

Measure of Degree of Influence.

$$DynRel\,(\mathbf{a}, \mathbf{z}) = \frac{|\mathcal{S}_{\text{bak}}(z) \cap \mathcal{S}_{\text{fwd}}(a)|}{|\mathcal{S}_{\text{fwd}}(a)|}$$

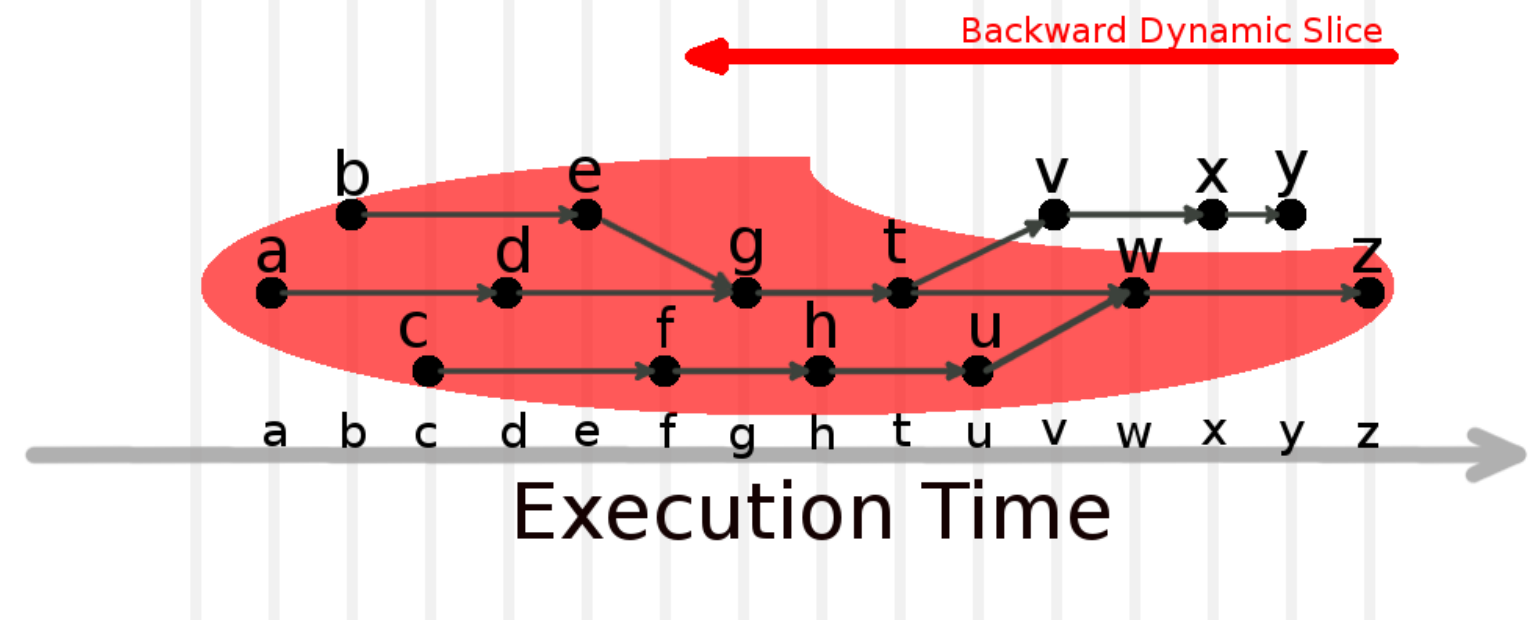


Dynamic Relevance

Measure of Degree of Influence.

$$\mathit{DynRel}\,(\mathbf{a}, \mathbf{z}) = \frac{|\mathcal{S}_{\mathrm{bak}}(z) \cap \mathcal{S}_{\mathrm{fwd}}(a)|}{|\mathcal{S}_{\mathrm{fwd}}(a)|}$$

- $\mathcal{S}_{\mathrm{bak}}(z)$: backward slice from (z);

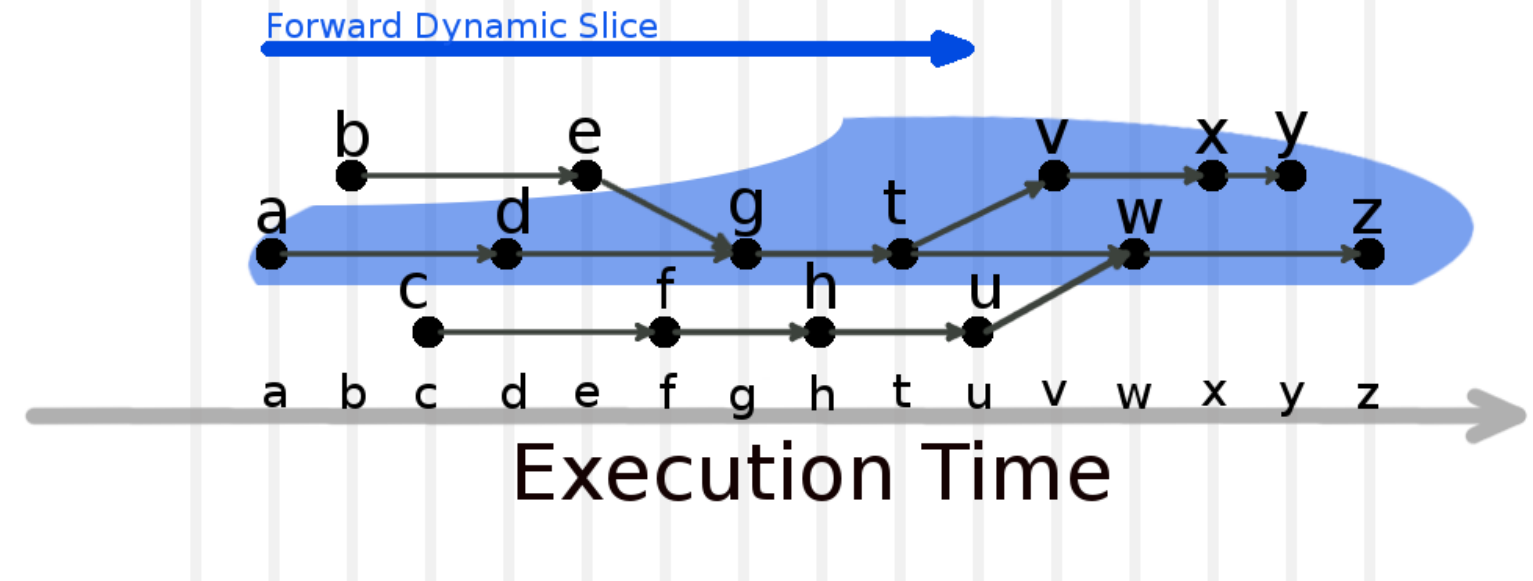


Dynamic Relevance

Measure of Degree of Influence.

$$DynRel\,(\mathbf{a}, \mathbf{z}) = \frac{|\mathcal{S}_{\text{bak}}(z) \cap \mathcal{S}_{\text{fwd}}(a)|}{|\mathcal{S}_{\text{fwd}}(a)|}$$

- $\mathcal{S}_{\text{bak}}(z)$: backward slice from (z);
- $\mathcal{S}_{\text{fwd}}(a)$: forward slice from (a);

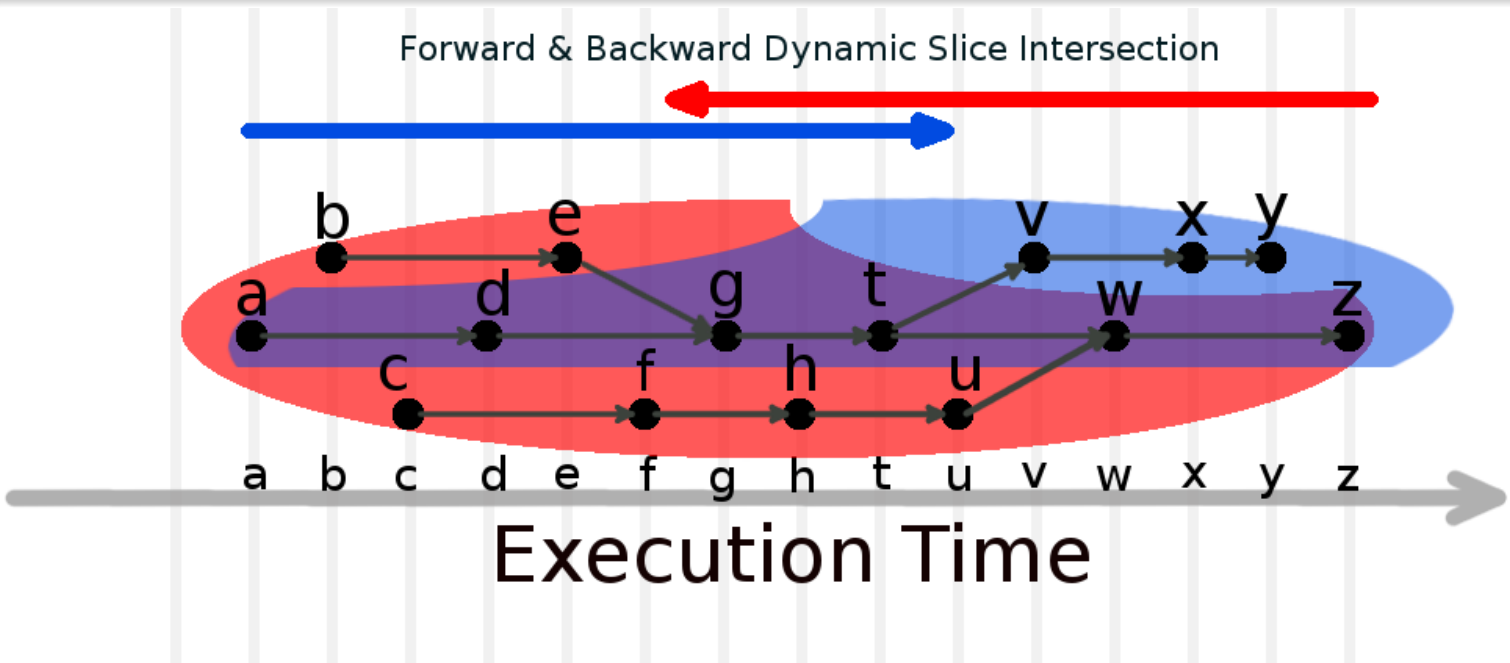


Dynamic Relevance

Measure of Degree of Influence.

$$DynRel\,(\mathbf{a}, \mathbf{z}) = \frac{|\mathcal{S}_{\text{bak}}(z) \cap \mathcal{S}_{\text{fwd}}(a)|}{|\mathcal{S}_{\text{fwd}}(a)|}$$

- $\mathcal{S}_{\text{bak}}(z)$: backward slice from (z);
- $\mathcal{S}_{\text{fwd}}(a)$: forward slice from (a);
- $\mathcal{S}_{\text{bak}}(z) \cap \mathcal{S}_{\text{fwd}}(a)$: runtime instructions common to $\mathcal{S}_{\text{bak}}(z)$ & $\mathcal{S}_{\text{fwd}}(a)$;

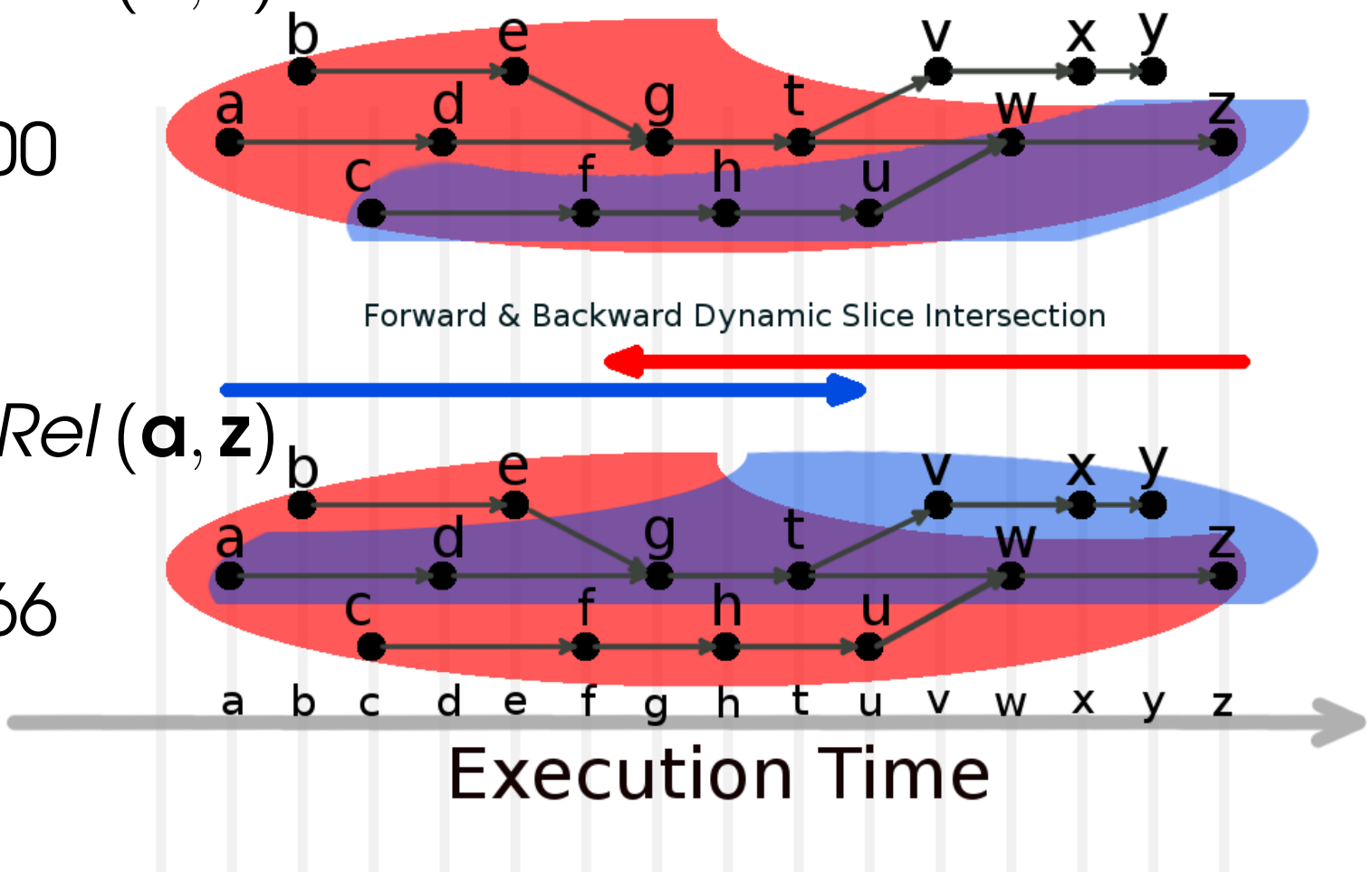


Dynamic Relevance

Measure of Degree of Influence.

$DynRel(\mathbf{c}, \mathbf{z}) = \frac{6}{6} = 1.00$

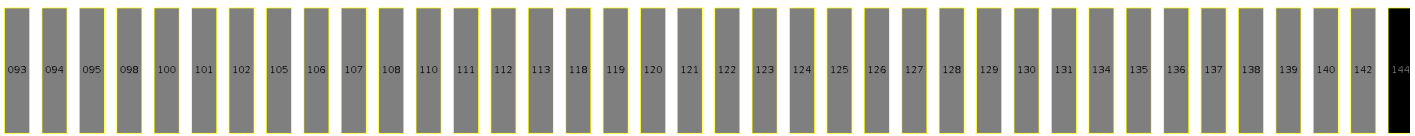
$DynRel(\mathbf{a}, \mathbf{z}) = \frac{6}{9} = 0.66$



Implementation: Looking Glass

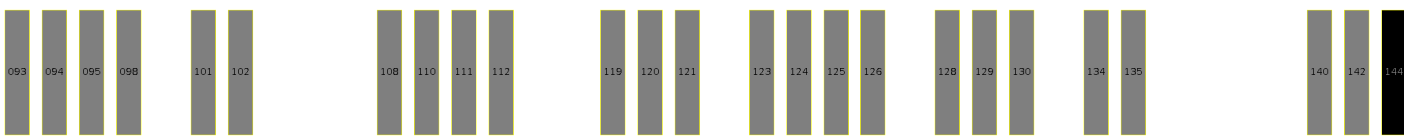
Program Execution Analyzer

Java Instrumenter & Execution Profiler



Program Execution Trace

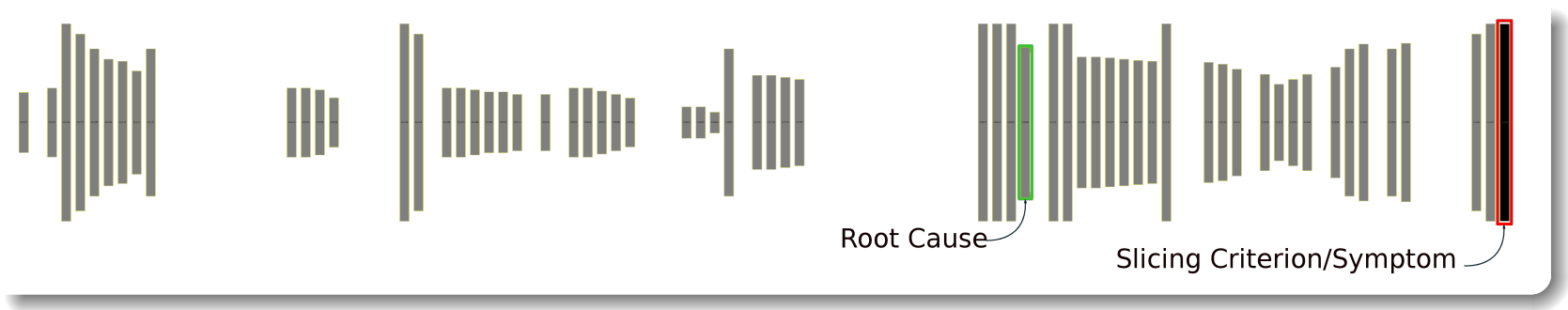
Execution Trace Analyzer & Slicer



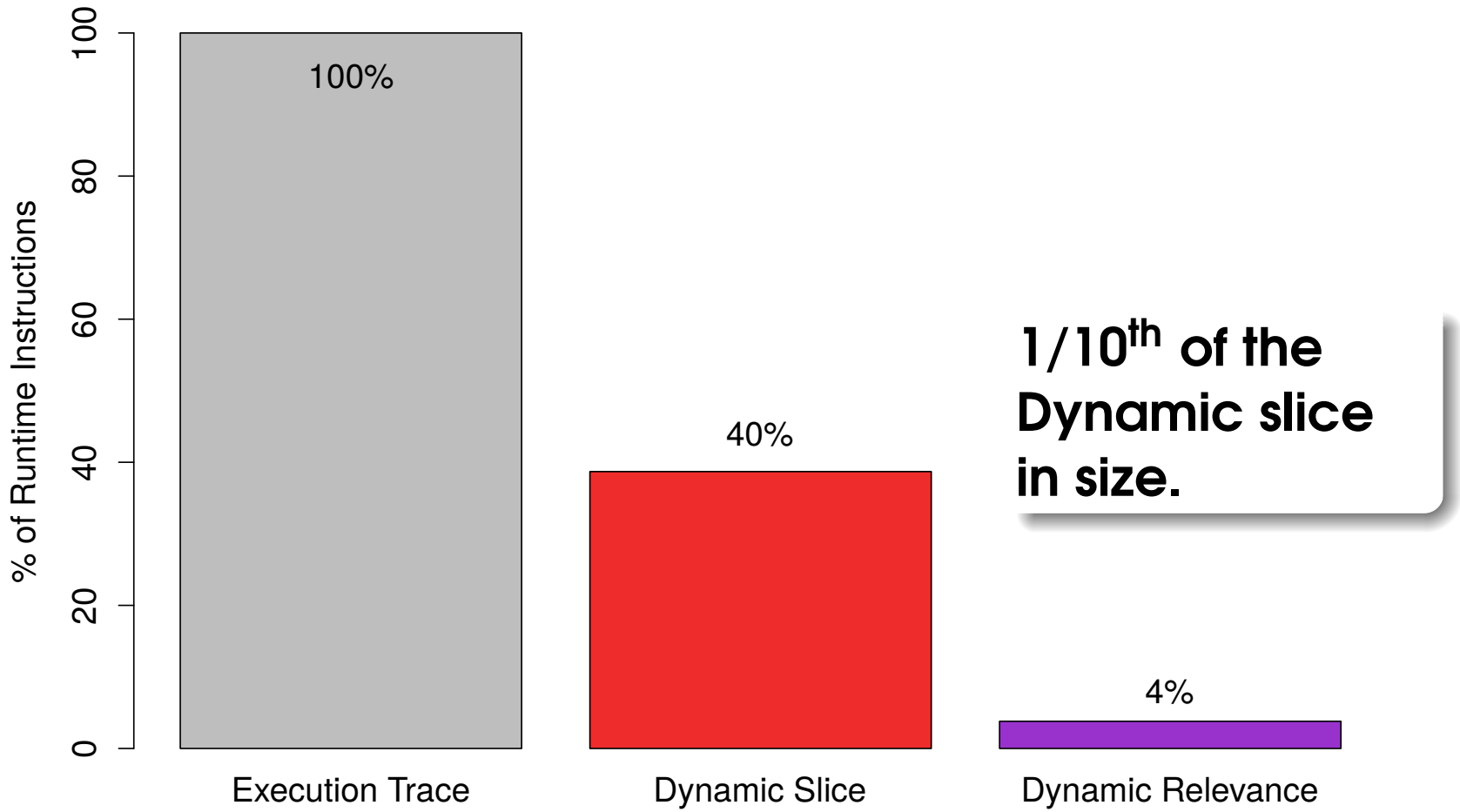
Backward and Forward Dynamic Slices

Emerging Results

- Task: Root Cause Analysis for an anomalous output.
- Subject: `NanoXML (>7,000 LOCs)`, XML Parser, Java
- 20 test executions



Emerging Results



Next Steps.

- Continuation of the empirical investigation.
- Development of novel software engineering analyses using discriminated influences.
- Envisioning variations of Dynamic Relevance.

Discriminating Influences among Instructions in a Dynamic Slice

Vijay Krishna Palepu James A. Jones

University of California, Irvine, USA