Object Flow Analysis – Taking an Object-centric View on Dynamic Analysis

Adrian Lienhard\textsuperscript{1}, Stéphane Ducasse\textsuperscript{2} and Tudor Gîrba\textsuperscript{1}

\textsuperscript{1}Software Composition Group, University of Bern, Switzerland
\textsuperscript{2}LISTIC, University of Savoie, France
A missing aspect of OO dynamic analysis

Typical characterization of OO program execution:

1) message passing
2) object interrelationships

How are objects passed through the system?
1) the method execution perspective

- MethodNode>>generate
- ASTTranslator class>>new
  - ...
  - ...
    - IRBuilder>>initialize
    - IRMethod class>>new
    - ...
  - ASTTranslator>>visitNode:
  - ASTTranslator>>ir
    - IRMethod>>compiledMethod
      - IRTranslator class>>new
I) the method execution perspective

“How is the IRMethod instance passed to MethodNode?”
2) the object interrelationship perspective
2) the object interrelationship perspective

“How is the IRMethod instance passed to MethodNode?”
Object Flow Analysis
track the transfer of object references
Object Flow Analysis

a) capture all references of an object

b) track the transfer of references

IRBuilder>>initialize
ir := IRMethod new.
...

IRBuilder>>startNewSequence
newSequence := IRSequence new.
newSequence method: ir.

IRSequence>>method: aMethod
method := aMethod.

ASTTranslator>>ir
^ builder ir.

MethodNode>>generate
ast := ASTTranslator new visitNode: self.
ir := ast ir.
^ ir compiledMethod.

IRMethod>>compiledMethod
^ compiledMethod := IRTranslator new
interpret: self;
compiledMethod.

IRTranslator>>interpret: ir
...
Object Flow Analysis

a) capture all references of an object

b) track the transfer of references

Object Flow Analysis — Taking an Object-centric View on Dynamic Analysis :: Adrian Lienhard :: lienhard@iam.unibe.ch
Object Flow Analysis

“How is the IRMethod instance passed to MethodNode?”
Application

*How do classes exchange objects?*
Inter-unit flow view

classes
  containedInPackage: 'AST-Nodes'
  mapTo: 'AST'.

classes
  hierarchyRootedIn: 'IRInstruction'
  mapTo: 'Intermediate-Representation'
Object Flow Analysis — Taking an Object-centric View on Dynamic Analysis

Adrian Lienhard
lienhard@iam.unibe.ch

Chronological propagation
<table>
<thead>
<tr>
<th>IRBuilder</th>
<th>timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRMethod</td>
<td></td>
</tr>
<tr>
<td>IRPop</td>
<td></td>
</tr>
<tr>
<td>IRReturn</td>
<td></td>
</tr>
<tr>
<td>IRSend</td>
<td></td>
</tr>
<tr>
<td>IRSequence</td>
<td></td>
</tr>
<tr>
<td>RBAssignmentNode</td>
<td></td>
</tr>
<tr>
<td>RBBBlockNode</td>
<td></td>
</tr>
<tr>
<td>RBMessageNode</td>
<td></td>
</tr>
</tbody>
</table>

- **Objects created in IRBuilder**
- **Objects passed out multiple times**
- **Objects passed through directly**
- **Objects stored in an instance variable**

---

Object Flow Analysis — Taking an Object-centric View on Dynamic Analysis :: Adrian Lienhard :: lienhard@iam.unibe.ch
Representing object flows

```
IRBuilder>>initialize
   ir := IRMethod new.
   ...

IRBuilder>>startNewSequence
   newSequence := IRSequence new.
   newSequence method: ir.

IRSequence>>method: aMethod
   method := aMethod.

ASTTranslator>>ir
   ^ builder ir.

MethodNode>>generate
   ast := ASTTranslator new visitNode: self.
   ir := ast ir.
   ir compiledMethod.
```

instantiation
   ↓
   field write
   ↓
   field read
   ↓
   argument
   ↓
   field write
   ↓
   return
   ↓
   return
   ↓
   ...
Representing object flows

```
IRBuilder>>initialize
   ir := IRMethod new.
   ...
IRBuilder>>startNewSequence
   newSequence := IRSequence new.
   newSequence method: ir.
IRSequence>>method: aMethod
   method := aMethod.
ASTTranslator>>ir
   ^ builder ir.
MethodNode>>generate
   ast := ASTTranslator new visitNode: self.
   ir := ast ir.
   ^ ir compiledMethod.
```

Diagram:
- Instance
- Alias
- Parent 0..1
- Child
- Instantiation
- Field write
- Field read
- Argument
- Field write
- Return
- Return
- ...
Representing object flows

instantiation
  ↓
 field write  
  ↓
field read  
  ↓
argument  
  ↓
field write  
  ↓
return  
  ↓
  ...

Instance

Alias

parent
  0..1
  ↓
child

ArgumentAlias

ReturnAlias

FieldWriteAlias

...
Representing object flows

Object Flow Analysis — Taking an Object-centric View on Dynamic Analysis :: Adrian Lienhard :: lienhard@iam.unibe.ch
Summary