## Towards Incremental Resource Usage Analysis

Elvira Albert¹, Jesús Correas¹, Germán Puebla², Guillermo Román-Díez²

(1) DSIC, Complutense University of Madrid (UCM), Spain  
(2) DLSIIS, Technical University of Madrid (UPM), Spain

### Resource Usage Analysis

#### Method Modification

- Infers the amount of resources that an execution will require
- Program analyzed from scratch

#### Incremental Resource Analysis

- Just analyzes the parts affected by the change
- Reuses computed information
- Takes care of propagating dependencies among affected methods
- Reconsitucts only affected components of Upper Bounds

### Incremental Inference of Pre-Analyses

1. Analyze (m2) using CP\textsubscript{m2}
2. If m\textsubscript{m4} \sqsubseteq CP\textsubscript{m4}
   - Use CP\textsubscript{m4} to analyze m4
3. If m\textsubscript{m5} \sqsubseteq CP\textsubscript{m5}
   - Use CP\textsubscript{m5} to analyze m5

### Incremental Inference of Upper Bounds

1. \[ UB_{in}(a) = 1 + 2 \cdot 4 + 3 \cdot (c - 1) \cdot a + 3 \cdot (a - 1) + 2 \cdot 3 \cdot (a - 5) \]
2. \[ UB_{in}(b) = 4 + 3 \cdot (4 \cdot 3 \cdot b - 1) \]
3. \[ UB_{in}(d) = 4 + 3 \cdot (d) \]

#### Upper Bounds

1. \[ UB_{m}, UB_{m2}, UB_{m4} \] cost expressions must be recomputed because they have been reanalyzed
2. \[ \psi \rightarrow m \rightarrow m2 \rightarrow m4 \rightarrow \psi \rightarrow m \rightarrow m5 \]
3. \[ UB_{m3}, UB_{m5} \] expressions are not recomputed
4. \[ UB_{m3} \] is maximized again to be inserted in UB\textsubscript{m}

### Experiments & Conclusions

#### Experiments

- Touching a method:
  - A method is modified but the modification does not affect its neighbours
- Adding method content:
  - A missing method is implemented
- Top-down development:
  - Simulates a top-down development process

#### Speedup

<table>
<thead>
<tr>
<th>Exp</th>
<th>Benchmark</th>
<th>Unweighted</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp</td>
<td>Speedup</td>
<td>W\textsubscript{op}</td>
<td>W\textsubscript{ir}</td>
</tr>
<tr>
<td>Tou</td>
<td>StringEncrypt</td>
<td>1.26</td>
<td>1.30</td>
</tr>
<tr>
<td>Tou</td>
<td>ParseTarHeader</td>
<td>1.54</td>
<td>1.30</td>
</tr>
<tr>
<td>Tou</td>
<td>TestDistance</td>
<td>1.38</td>
<td>1.80</td>
</tr>
<tr>
<td>Dev</td>
<td>StringEncrypt</td>
<td>1.35</td>
<td>1.31</td>
</tr>
<tr>
<td>Dev</td>
<td>ParseTarHeader</td>
<td>1.29</td>
<td>1.26</td>
</tr>
<tr>
<td>Dev</td>
<td>TestDistance</td>
<td>1.36</td>
<td>2.26</td>
</tr>
</tbody>
</table>

#### Conclusions

- All main steps of a Resource Usage Analysis are handled:
  - OO Pre-Analyses
  - Recomputing Upper Bounds
- Only those components affected by the change are reconstructed
- Feasible and efficient
- Promising experimental results
- To appear in PEP’12