

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

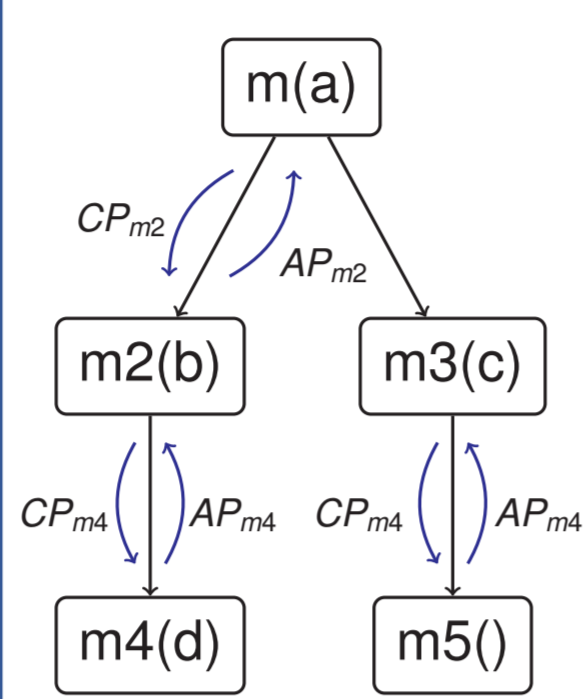
Resource Analysis

- 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
- Reconstructs only affected components of Upper Bounds

Resource Usage Analysis



| Method | Dom | Summaries ($CP \mapsto AP$) |
|--------|------|--------------------------------------|
| m(a) | sign | $\{a = \perp\} \mapsto \perp$ |
| | size | $\top \mapsto \perp$ |
| m2(b) | sign | $\{b = \perp\} \mapsto \{r = 0\}$ |
| | size | $\top \mapsto \{r = 0\}$ |
| m3(c) | sign | $\{c = \perp\} \mapsto \perp$ |
| | size | $\top \mapsto \top$ |
| m4(d) | sign | $\{d = \perp\} \mapsto \{r = \top\}$ |
| | size | $\{d > 0\} \mapsto \{r = d + 5\}$ |
| m5() | sign | $\perp \mapsto \{r > 0\}$ |
| | size | $\perp \mapsto \{r = 10\}$ |

CP: Description in the abstract domain referring to input vars
AP: Description in the abstract domain referring to output vars

$$UB_m(a) = 1 + 2 + 4 + 3 * (a + 1) + 2 + 3 * (a) + 5$$

$$UB_{m2}(b) = 2 + 4 + 3 * (b + 1)$$

$$UB_{m3}(c) = 2 + 3 * (c) + 5$$

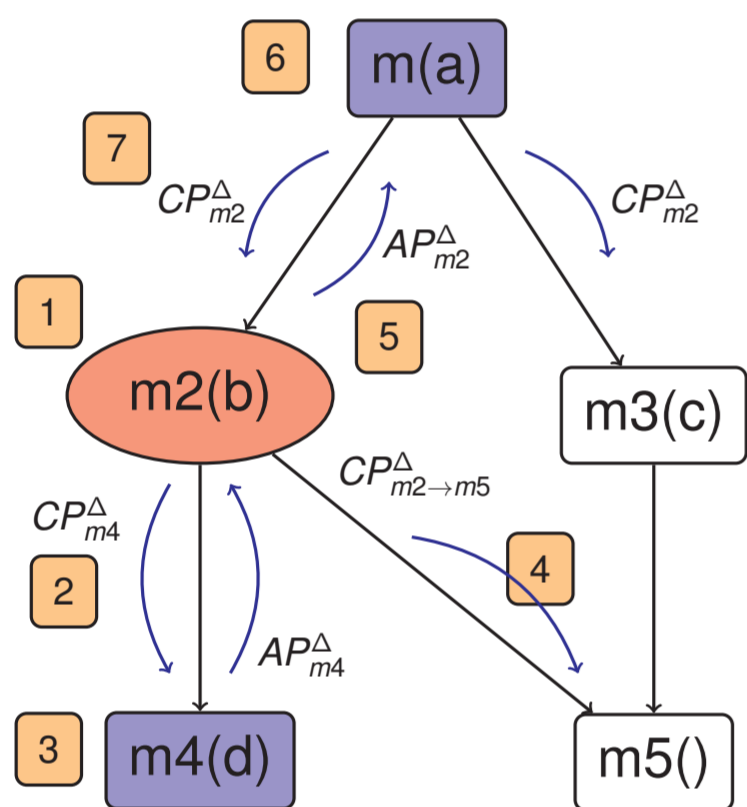
$$UB_{m4}(d) = 4 + 3 * (d)$$

$$UB_{m5}() = 5$$

$\varphi_{m \rightarrow m2}$: size abstractions & size analysis from m to $m2$ (computed in summary)
 $\text{maxim}(\varphi)$: maximize the cost using φ . Example:
for ($i=1..n$) $C_{outer} = n * C_{inner}$
for ($j=i..m$) $\varphi = \{i \leq j \leq m\} \rightarrow C_{outer} = n * m$

Incremental Inference of Pre-Analyses

Method Modification



Descendants

- Analyze ($m2$) using CP_{m2}
- Is $CP_{m4}^\Delta \sqsubseteq CP_{m4}$?
 $\sqsubseteq \rightarrow$ Reuse AP_{m4}
 $\not\sqsubseteq \rightarrow$, analyze $m4$ using $CP_{m4} \sqcup CP_{m4}^\Delta$
- If $\not\sqsubseteq$ Analyze $m4(d)$
- Is $CP_{m2 \rightarrow m5}^\Delta \sqsubseteq CP_{m5}$?
 $\sqsubseteq \rightarrow$ Use AP_{m5}
 $\not\sqsubseteq \rightarrow$ Analyze $m5$ using $CP_{m2 \rightarrow m5}^\Delta \sqcup CP_{m5}$

Ascendants

- Is $AP_{m2}^\Delta \sqsubseteq AP_{m2}$?
 $\sqsubseteq \rightarrow$ Do nothing
 $\not\sqsubseteq \rightarrow$ Analyze m
- Analyze m as for descendants
- Is $CP_{m2}^\Delta \sqsubseteq CP_{m2}$?
 $\sqsubseteq \rightarrow$ Reuse AP_{m2}
 $\not\sqsubseteq \rightarrow$ Analyze $m2$ again using $CP_{m \rightarrow m2}^\Delta \sqcup CP_{m2}$

Incremental Inference of Upper Bounds

Updating Upper Bounds

$$UB_m(a) = 1 + 2 + 5 * (4 + 3 * (a - 1) * (a)) + 5 + 2 + 3 * (a) + 5$$

$$UB_{m2}(b) = 2 + 5 * ((4 + 3 * (b - 1)) * (b)) + 5$$

$$UB_{m3}(c) = 2 + 3 * (c) + 5$$

$$UB_{m4}(d) = 4 + 3 * (d)$$

$$UB_{m5}() = 5$$

Upper Bounds

- UB_m, UB_{m2}, UB_{m4} cost expressions must be recomputed because they have been reanalyzed
- $\varphi_{m \rightarrow m2}, \varphi_{m2 \rightarrow m4}, \varphi_{m \rightarrow m3}$ may have changed \rightarrow we need to maximize the cost expressions
- UB_{m3}, UB_{m5} expressions are not recomputed
- UB_{m3} is maximized again to be inserted in UB_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

| Exp | Benchmark | Speedup | | | | | |
|-----|----------------|------------|----------|-------|-----------|----------|-------|
| | | Unweighted | | | Weighted | | |
| | | S_{CRs} | S_{UB} | S_T | W_{CRs} | W_{UB} | W_T |
| Tou | StringEncrypt | 10.90 | 7.09 | 8.11 | 7.56 | 3.92 | 4.74 |
| Tou | ParseTarHeader | 5.52 | 2.03 | 2.54 | 8.09 | 3.31 | 4.07 |
| Tou | TestDistance | 3.09 | 4.95 | 3.26 | 4.05 | 6.09 | 4.25 |
| Add | StringEncrypt | 1.26 | 1.30 | 1.28 | 2.04 | 2.40 | 2.26 |
| Add | ParseTarHeader | 1.54 | 1.30 | 1.37 | 2.46 | 2.35 | 2.39 |
| Add | TestDistance | 1.38 | 1.80 | 1.43 | 1.90 | 2.44 | 1.96 |
| Dev | StringEncrypt | 1.35 | 1.31 | 1.32 | 1.51 | 1.63 | 1.58 |
| Dev | ParseTarHeader | 1.29 | 1.26 | 1.27 | 1.29 | 1.41 | 1.37 |
| Dev | TestDistance | 1.36 | 2.26 | 1.44 | 1.39 | 2.16 | 1.46 |

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 PEPM'12