

Cost Analysis Summary II

Monday, July 18, 2022 9:25 AM

T

Static Bound Inference Analysis.

- 1 Overlapped with Bound Analysis. Worst Case Runtime analysis

WCET on systems: \cite{}

[GustafssonEL05] Towards a Flow Analysis for Embedded System C Programs

--> abstract interpretation.

--> on embedded system of c program

[AlbertAGP08] Automatic Inference of Upper Bounds for Recurrence Relations in Cost Analysis

--> invariant generation through ranking functions

General While language:

[BrockschmidtEFFG16]

Analyzing Runtime and Size Complexity of Integer Programs

--> invariant generation through ranking functions

[AliasDFG10] Multi-dimensional Rankings, Program Termination, and Complexity Bounds of Flowchart Programs

--> invariant generation through ranking functions

[Flores-MontoyaH14] Resource Analysis of Complex Programs with Cost Equations

--> invariant generation through cost equations or ranking functions

[GulwaniJK09] Control-flow Refinement and Progress Invariants for Bound Analysis

--> program abstraction and invariant inference

[] Bound Analysis using Backward Symbolic Execution

--> program abstraction and invariant inference

- ① 1997. Thomas. via graph Reachability.

- ② 2008. Upper Bound for Recurrence Relation in Cost Analysis.

⇒ ① generate RR (recurrence Relation).

★ ② solve how to compute RR into closed form. → focus on solve the closed form

⇒ by ranking function & loop invariant & partial evaluation

- ③ → 2008. Numerical Abstract Domain, (on Expression Abstraction & Max Operator) apply in Timing analysis.

⇒ exp abstraction + linear Relation. & Max

⇒ generate abstract expressions S → programmer provide / symbolic execution

⇒ abstract interpretation

↳ exp abstraction

↳ linear relational abstract domain + disjunctive reasoning. ↳ with max.

★ set of inference rules for computing bounds.

eg: $z_x \leq c \Rightarrow z_{sx} \leq \omega^c$

tree $\Rightarrow z_{\text{left}} = \max(z_x, -z_x)$

$\sum a_i z_{xi} = a \Rightarrow \sum a_i z_{xy} = a z_y$

$z_x \leq z_y + c \Rightarrow z_{sx} \leq \omega^c \times z_y$.

... ↓ computation algorithm,

- ④ 2009. - Control flow refinement. more detail in RB analysis note

1. refine if-branch.

↓ 2. still solve Bound for outside loop by conjunction of multiple path.

- ⑤ 2009. SPEED

↓ Extend to inter-procedure with implementation improvement.

- ⑥ 2009: Symbolic Execution

similar to SPEED.

- ⑦ 2010. Reachability Bound Problem. ↳ detail & Example in RB.

⇒ by inferred closed loop Bound for Location Reachability. ⇒ not for location.

⇒ entire cost count the max

- ⑧ 2012. Bound

⇒ the same as the PLDI'17. Complexity ...

solve Bound for each SCC. by assigning each edge with an abs var.

⇒ compute Bound of Dec var - use it as Bound of Whole SCC

⇒ Bound of edge.

① still cannot solve multiBound odd.

② ⇒ for entire cost. still take the max.

- ⇒ 2013. Size-Change Abstraction & Max Plus automata

Technique paper on how to solve equation to infer Bound

①. automatic ↳ characterizing the asymptotic complexity bounds obtained by size-change.

②. size-change predicate (SCP),

$SCT = \bigvee P(SCP), \text{ SCS } \in P(SCT)$

$a, b \models x \approx y \text{ if } 0 \leq a \leq b$ holds

$a, b \rightarrow D, N$

a bit far from solving Bound.

- ⑨ 2014. Resource Analysis with Cost Equation.

⇒ by infer Loop Bound & Estimate the Worst Case

- ⑩ 2014: Scalable Static Analysis for Bound Analysis and Amortized Complexity Analysis.

efficiently improvement in solving the Bound

But still estimate the Worst Case cost Consumption. for program

- ⑪ 2016: Running time & size Complexity of Integer Programs.

⇒ infer Bound for variables, ⇒ then apply to program cost infer.

⇒ But still count entire Worst Case cost.

- ⑫ 2017: Sinn: Complexity & Resource Bound analysis of Imperative Program Using DC

⇒ ① doesn't consider path sensitive

⇒ still worst case.