

Saturn

Motivation

check temporal safety properties

- Precise
- Interprocedural
- Scale ($\approx$ linux kernel)

How it works

Boolean formulas for each bit

Integers $\psi \vdash e \stackrel{F}{\Rightarrow} [b_{n-1}, \dots, b_0]_S$

$\psi \vdash e' \stackrel{F}{\Rightarrow} [b'_{n-1}, \dots, b'_0]_S$

$\psi \vdash e \text{ and } e' \stackrel{F}{\Rightarrow} [b_{n-1} \wedge b'_{n-1}, \dots, b_0 \wedge b'_0]_S$

Statements and path sensitivity

Use guards $\psi \vdash c \stackrel{S}{\Rightarrow} b$

$G, \psi \vdash \text{assume } c \stackrel{S}{\Rightarrow} \langle G \wedge b, \psi \rangle$

Merge paths with merge scalar which moves guards into boolean formulas.

Pointers and structures also modelled.

Function summaries

lock $l \leftarrow$ add state

$S = \{\text{error}, s_1, \dots, s_n\}$

$\Sigma = \{P_{in}, P_{out}, M, R\}$

$R \subseteq \Sigma^{in} \times 2^{|P_{in}|} \times S \times 2^{|P_{out}|} \times S$

Generated it supplied S, P_{in}, P_{out} .

$R = \{(s_{in}, \text{incond}, \text{outcond}, s, s') \mid \text{is_satisfiable}$

$(V_0(s_{in}=s) \wedge V_0(\text{incond}) \wedge V(s_{in}=s') \wedge V(\text{outcond}))\}$

$M =$ has object it changed in method.

Classification

Unsound and incomplete

- Loop unrolling 2 times
- Aliasing

Expressiveness

Safety properties

Can check assertions

Modularity

To bit level $\Leftarrow$ Intraprocedural

Function summaries $\Leftarrow$ Interprocedural

Makes lightweight it using prebuilt checkers

Scalability

20 hours = Linux kernel

50 min in cluster