Automatic Abstraction for Congruences

A Story of Beauty and the Beast

Andy King and Harald Søndergaard

Portcullis Computer Security

University of Melbourne

Structure of this talk

- Related work:
  - Philippe Granger: Static Analysis of Linear Congruence Equalities among Variables of a Program, TAPSOFT, 1991!
  - Markus Müller-Olm and Helmut Seidl: Analysis of Modular Arithmetic, TOPLAS, 2007
  - David Monniaux: Automatic Modular Abstractions for Linear Constraints, POPL, 2009
  - Björn Wachter and Lijun Zhang: Best Probabilistic Transformers, VMCAI, 2010?
- Heart of the technique:
  - describe (abstract) a Boolean function with a system of congruences
  - technique interleaves SAT solving and merge for congruences

Describing a function with congruence constraints

- Consider the function $f = (c_0' \oplus c_0) \wedge (c_1' \leftrightarrow (c_1 \oplus c_0))$ where $\oplus$ denotes exclusive or.
- Observe that f is described by the congruence constraints:

$$s = \left\{ \begin{array}{r} c_0 + c_0' \equiv_4 1 \\ 2c_1 + 2c_0' + 2c_1' \equiv_4 2 \end{array} \right\} = \left\{ \begin{array}{c} c_0 + c_0' \equiv_4 1 \\ c_0 + 2c_1 + 1 \equiv_4 c_0' + 2c_1' \end{array} \right\}$$

- s describes f since every solution of f is a solution of s:

c_0	c_1	c_0'	c_1'	$c_0 + c_0' \equiv_4 1$	$2c_1 + 2c_0' + 2c_1' \equiv_4 2$	s	f
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	1	1	1
0	0	1	1	1	0	0	0
0	1	0	0	0	1	0	0
0	1	0	1	0	0	0	0
0	1	1	0	1	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Finding a congruence system s that describes f

1.1 Find a solution to f, namely, the satisfying assignment:

$$m_1 = \{c_0 \mapsto 1, c_1 \mapsto 0, c_0' \mapsto 0, c_1' \mapsto 1\}$$

1.2 Represent m_1 as a triangular system of congruences:

$$s_1 = \left\{ \begin{array}{llllll} c_0 & & & & \equiv_4 1 & \\ & c_1 & & & \equiv_4 0 & \\ & & c_0' & & \equiv_4 0 & \\ & & & c_1' & \equiv_4 1 & \leftarrow \end{array} \right\}$$

1.3 Construct a function g_1 that holds iff $c_1' \equiv_4 1$ does not hold:

$$g_1 = (c_1' \oplus 1)$$

Finding a congruence system s that describes f (cont')

2.1 Find a solution to $f \wedge g_1$, namely, the satisfying assignment:

$$m_2 = \{c_0 \mapsto 1, c_1 \mapsto 1, c_0' \mapsto 0, c_1' \mapsto 0\}$$

2.2 Represent m_2 as a system of congruence constraints s_2':

$$s_2' = \left\{ \begin{array}{l} c_0 \equiv_4 1 \\ c_1 \equiv_4 1 \\ c_0' \equiv_4 0 \\ c_1' \equiv_4 0 \end{array} \right\} \quad \text{and recall } s_1 = \left\{ \begin{array}{l} c_0 \equiv_4 1 \\ c_1 \equiv_4 0 \\ c_0' \equiv_4 0 \\ c_1' \equiv_4 1 \end{array} \right\}$$

2.3 Merge s_2' and s_1 to obtain $s_2 = \left\{ \begin{array}{lllll} c_0 & & & & \equiv_4 1 \\ & c_1 & & + c_1' & \equiv_4 1 \\ & & c_0' & & \equiv_4 0 \leftarrow \end{array} \right\}$

2.4 Construct a function g_2 that holds iff $c_0' \equiv_4 0$ does not hold:

$$g_2 = (c_0' \oplus 0)$$

Finding a congruence system s that describes f (cont')

3.1 Find a solution to $f \wedge g_2$, namely, the satisfying assignment:

$$m_3 = \{c_0 \mapsto 0, c_1 \mapsto 0, c'_0 \mapsto 1, c'_1 \mapsto 0\}$$

3.2 Represent m_3 as a system of congruence constraints s'_3:

$$s'_3 = \left\{ \begin{array}{l} c_0 \equiv_4 0 \\ c_1 \equiv_4 0 \\ c'_0 \equiv_4 1 \\ c'_1 \equiv_4 0 \end{array} \right\} \qquad \text{and recall } s_2 = \left\{ \begin{array}{r} c_0 \equiv_4 1 \\ c_1 + c'_1 \equiv_4 1 \\ c'_0 \equiv_4 0 \end{array} \right\}$$

3.3 Merge s'_3 and s_2 to obtain

$$s_3 = \left\{ \begin{array}{lllll} c_0 & & + c'_0 & & \equiv_4 1 \\ & c_1 & + c'_0 & + c'_1 & \equiv_4 1 \leftarrow \end{array} \right\}$$

3.4 Construct g_3 that holds iff $c_1 + c'_0 + c'_1 \equiv_4 1$ does not hold:

$$\begin{array}{rlcll} g_3 = & (t_0 \leftrightarrow c_1 \oplus c'_0) & \wedge & (t_1 \leftrightarrow c_1 \wedge c'_0) & \wedge \\ & (t'_0 \leftrightarrow c'_1 \oplus t_0) & \wedge & (t'_1 \leftrightarrow (c'_1 \wedge t_0) \oplus t_1) & \wedge \\ & (\neg t'_0 \vee t'_1) & & & \end{array}$$

Finding a congruence system s that describes f (cont')

4.1 Find a solution to $f \wedge g_3$, namely, the satisfying assignment:

$$m_4 = \{c_0 \mapsto 0, c_1 \mapsto 1, c_0' \mapsto 1, c_1' \mapsto 1\}$$

4.2 Represent m_4 as a system of congruence constraints s_4':

$$s_4' = \left\{ \begin{array}{l} c_0 \equiv_4 0 \\ c_1 \equiv_4 0 \\ c_0' \equiv_4 1 \\ c_1' \equiv_4 0 \end{array} \right\}$$

4.3 Merge s_3 and s_4' to obtain s_4:

$$s_4 = \left\{ \begin{array}{lllll} c_0 & & + c_0' & & \equiv_4 1 \\ & 2c_1 & + 2c_0' & + 2c_1' & \equiv_4 2 \leftarrow \end{array} \right\}$$

4.4 Construct g_4 that holds iff $c_1 + c_0' + c_1' \equiv_2 1$ does not hold:

$$g_4 = (c_1 \oplus c_0' \oplus c_1') \oplus 1$$

Finding a congruence system s that describes f (cont')

5.1 Detect that $f \wedge g_4$ does not have a solution

$$s_4 = \left\{ \begin{array}{lllll} c_0 & & + c'_0 & & \equiv_4 1 \quad \leftarrow \\ & 2c_1 & + 2c'_0 & + 2c'_1 & \equiv_4 2 \quad \surd \end{array} \right\}$$

5.2 Construct g_5 that holds iff $c_0 + c'_0 \equiv_4 1$ does not hold:

$$g_5 = (t_0 \leftrightarrow c_1 \oplus c'_0) \wedge (t_1 \leftrightarrow c_1 \wedge c'_0) \wedge (\neg t_0 \vee t_1)$$

5.3 Detect that $f \wedge g_5$ does not have a solution

$$s_4 = \left\{ \begin{array}{lllll} c_0 & & + c'_0 & & \equiv_4 1 \quad \surd \\ & 2c_1 & + 2c'_0 & + 2c'_1 & \equiv_4 2 \quad \surd \end{array} \right\}$$

Summary of story so far

- The systems s_1, s_2, s_3, s_4 constitute an increasing chain of congruence constraints:
  - The system s_{i+1} has strictly more solutions than s_i;
  - The maximal number of systems in chain is $pn + 1$ where $2^p = 4$ is the modulo [TOPLAS, 2007]
- But does it scale?
  - Depends on hardness of the SAT/SMT instance
  - Depends on the join algorithm (needs to be inplace)

Parity example

ℓ_0: $p := 0;\ y := x;$
ℓ_1: while $(y \neq 0)$
$\quad\quad y := y \,\&\, (y - 1);$
$\quad\quad p := 1 - p;$
ℓ_2: skip

Then $t'_1 = \langle \ell_0, \ell_1, c_1 \rangle$, $t'_2 = \langle \ell_1, \ell_1, c_2 \rangle$ and $t'_3 = \langle \ell_1, \ell_2, c_3 \rangle$ where

$$c_1 = \begin{cases} (\wedge_{i=0}^{15} p'_i \equiv_2 0) \;\wedge \\ (\wedge_{i=0}^{15} y'_i \equiv_2 x_i) \wedge \\ (\wedge_{i=0}^{15} x'_i \equiv_2 x_i) \end{cases}$$

$$c_2 = \begin{cases} p_0 + p'_0 \equiv_2 1 \quad \wedge \\ (\wedge_{i=1}^{15} p_i \equiv_2 p'_i) \wedge \\ (\wedge_{i=0}^{15} x_i \equiv_2 x'_i) \wedge \\ y'_0 \equiv_2 0 \quad\quad\quad \wedge \\ 1 + \sum_{i=1}^{15} y'_i \equiv_2 \sum_{i=0}^{15} y_i \end{cases}
\qquad
c_3 = \begin{cases} (\wedge_{i=0}^{15} p'_i \equiv_2 p_i) \wedge \\ (\wedge_{i=0}^{15} x'_i \equiv_2 x_i) \wedge \\ (\wedge_{i=0}^{15} y_i \equiv_2 0) \;\; \wedge \\ (\wedge_{i=0}^{15} y'_i \equiv_2 0) \end{cases}$$

Bit reversal example

$$
\begin{array}{lll}
\ell_0\colon & y := x; & \\
& y := ((y \gg 1) \ \& \ \mathrm{0x5555}) & \mid ((y \ \& \ \mathrm{0x5555}) \ll 1); \\
& y := ((y \gg 2) \ \& \ \mathrm{0x3333}) & \mid ((y \ \& \ \mathrm{0x3333}) \ll 2); \\
& y := ((y \gg 4) \ \& \ \mathrm{0x0F0F}) & \mid ((y \ \& \ \mathrm{0x0F0F}) \ll 4); \\
& y := (y \gg 8) & \mid (y \ll 8); \\
\ell_1\colon & \mathrm{skip} &
\end{array}
$$

Then $t' = \langle \ell_0, \ell_1, c \rangle$ where $c = \bigwedge_{i=0}^{15} (x'_i \equiv_{2^{16}} x_i \wedge y'_{15-i} \equiv_{2^{16}} x_i)$

Bonus tracks

- Presented a new algorithm for congruence closure;
- Show how inequalities can be traced (see paper);

$$
\begin{array}{ll}
\ell_0: & \text{assume}(0 < n); \\
 & x := 0;\ y := 0; \\
\ell_1: & \text{while } (x < n) \\
 & \qquad y := y + 2; \\
 & \qquad x := x + 1; \\
\ell_2: & \text{skip}
\end{array}
\quad \Rightarrow \quad
\begin{array}{ll}
\ell_0: & \text{assume}(0 < n); \\
 & x := 0;\ y := 0;\ \delta := n - x; \\
\ell_1: & \text{while } (0 < \delta) \\
 & \qquad y := y + 2; \\
 & \qquad x := x + 1; \\
 & \qquad \delta := n - x; \\
\ell_2: & \text{skip}
\end{array}
$$

- Formulate the ideas with an unrestricted flowchart language with non-linear, bit-manipulating operations (see paper)