CASCO: Contract Aware Secure COmpilation ???

Marco Guarnieri
IMDEA Software Institute

Marco Patrignani
University of Trento

Matteo Possamai
Unknown...

Basic Types

(Registers)	x	$\in$	Regs
(Values)	n, ℓ	$\in$	$\mathit{Vals} = \mathbb{N} \cup \{\bot\}$

Syntax

(Expressions)	e	$:=$	$n \mid x \mid \ominus e \mid e_1 \otimes e_2$
(Instructions)	i	$:=$	$\mathbf{skip} \mid x \leftarrow e \mid \mathbf{load}\ x, e \mid$
			$\mathbf{store}\ x, e \mid \mathbf{jmp}\ e \mid \mathbf{beqz}\ x, \ell \mid$
			$x \xleftarrow{e'?} e \mid \mathbf{spbarr}$
(Programs)	p	$:=$	$n : i \mid p_1; p_2$

Fig. 1. μASM Syntax

Expression evaluation

$$[\![n]\!](a) = n \qquad [\![x]\!](a) = a(x) \qquad [\![\ominus e]\!](a) = \ominus [\![e]\!](a) \qquad [\![e_1 \otimes e_2]\!](a) = [\![e_1]\!](a) \otimes [\![e_2]\!](a)$$

Instruction evaluation

SKIP

$$\frac{p(a(\mathbf{pc})) = \mathbf{skip}}{\langle m, a\rangle \to \langle m, a[\mathbf{pc} \mapsto a(\mathbf{pc}) + 1]\rangle}$$

BARRIER

$$\frac{p(a(\mathbf{pc})) = \mathbf{spbarr}}{\langle m, a\rangle \to \langle m, a[\mathbf{pc} \mapsto a(\mathbf{pc}) + 1]\rangle}$$

ASSIGN

$$\frac{p(a(\mathbf{pc})) = x \leftarrow e \qquad x \neq \mathbf{pc}}{\langle m, a\rangle \to \langle m, a[\mathbf{pc} \mapsto a(\mathbf{pc}) + 1, x \mapsto [\![e]\!](a)]\rangle}$$

CONDITIONALUPDATE-SAT

$$\frac{p(a(\mathbf{pc})) = x \xleftarrow{e'?} e \qquad [\![e']\!](a) = 0 \qquad x \neq \mathbf{pc}}{\langle m, a\rangle \to \langle m, a[\mathbf{pc} \mapsto a(\mathbf{pc}) + 1, x \mapsto [\![e]\!](a)]\rangle}$$

CONDITIONALUPDATE-UNSAT

$$\frac{p(a(\mathbf{pc})) = x \xleftarrow{e'?} e \qquad [\![e']\!](a) \neq 0 \qquad x \neq \mathbf{pc}}{\langle m, a\rangle \to \langle m, a[\mathbf{pc} \mapsto a(\mathbf{pc}) + 1]\rangle}$$

TERMINATE

$$\frac{p(a(\mathbf{pc})) = \bot}{\langle m, a\rangle \to \langle m, a[\mathbf{pc} \mapsto \bot]\rangle}$$

LOAD

$$\frac{p(a(\mathbf{pc})) = \mathbf{load}\ x, e \qquad x \neq \mathbf{pc} \qquad n = [\![e]\!](a)}{\langle m, a\rangle \xrightarrow{\texttt{load } n} \langle m, a[\mathbf{pc} \mapsto a(\mathbf{pc}) + 1, x \mapsto m(n)]\rangle}$$

STORE

$$\frac{p(a(\mathbf{pc})) = \mathbf{store}\ x, e \qquad n = [\![e]\!](a)}{\langle m, a\rangle \xrightarrow{\texttt{store } n} \langle m[n \mapsto a(x)], a[\mathbf{pc} \mapsto a(\mathbf{pc}) + 1]\rangle}$$

BEQZ-SAT

$$\frac{p(a(\mathbf{pc})) = \mathbf{beqz}\ x, \ell \qquad a(x) = 0}{\langle m, a\rangle \xrightarrow{\texttt{pc } \ell} \langle m, a[\mathbf{pc} \mapsto \ell]\rangle}$$

BEQZ-UNSAT

$$\frac{p(a(\mathbf{pc})) = \mathbf{beqz}\ x, \ell \qquad a(x) \neq 0}{\langle m, a\rangle \xrightarrow{\texttt{pc } a(\mathbf{pc})+1} \langle m, a[\mathbf{pc} \mapsto a(\mathbf{pc}) + 1]\rangle}$$

JMP

$$\frac{p(a(\mathbf{pc})) = \mathbf{jmp}\ e \qquad \ell = [\![e]\!](a)}{\langle m, a\rangle \xrightarrow{\texttt{pc } \ell} \langle m, a[\mathbf{pc} \mapsto \ell]\rangle}$$

Fig. 2. μASM semantics for a program p