A Formal Equational Theory for Call-By-Push-Value

Christine Rizkallah

Dmitri Garbuzov – Steve Zdancewic

University of Pennsylvania

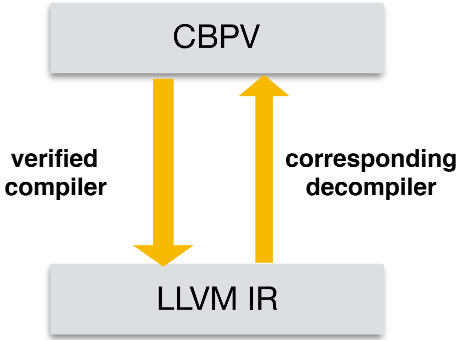
Dagstuhl Seminar: Secure Compilation – 16 May 2018

Recap Vellvm: Verifying the LLVM IR

- a formal semantics allows reasoning about LLVM code
- previously: LLVM IR formalization in Coq with large monolithic proofs (unmaintainable)
- aim: build a **maintainable** framework for reasoning about LLVM programs and LLVM transformations in Coq.

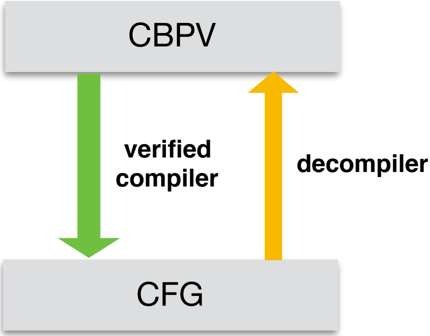
Recap: Our Plan

Optimizations (existing LLVM) $\subseteq$ Equational Theory $\subseteq$ Contextual Equiv.



Our Progress

Optimizations $\subseteq$ Equational Theory $\subseteq$ Contextual Equiv.



Contextual Equivalence

- contextual equivalence is gold standard for program equiv.
- programs "behave similarly" under **any context**

Contextual Equivalence

- contextual equivalence is gold standard for program equiv.
- programs "behave similarly" under **any context**
- contextually equivalent: **co-terminate in any context**
  - co-terminate: both terminate or both diverge

Soundness

- sound relation only relates contextually equiv. programs

Soundness

- sound relation only relates contextually equiv. programs
- sound relation = congruence $\wedge$ adequate

Soundness

- sound relation only relates contextually equiv. programs
- sound relation = congruence $\wedge$ adequate
  - adequate relation $\subseteq$ co-terminate

Soundness

- sound relation only relates contextually equiv. programs
- sound relation = congruence $\wedge$ adequate
  - adequate relation $\subseteq$ co-terminate
  - congruence = equivalence $\wedge$ compatible

Soundness

- sound relation only relates contextually equiv. programs
- sound relation = congruence $\wedge$ adequate
  - adequate relation $\subseteq$ co-terminate
  - congruence = equivalence $\wedge$ compatible

Defined an equational theory and proved it is sound

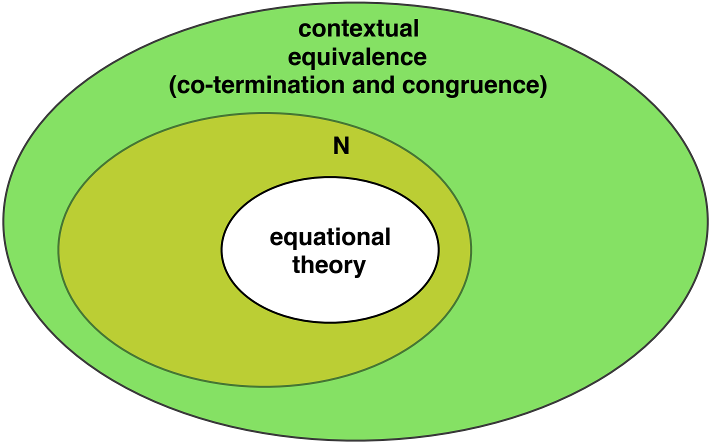
Equational Theory

- Defined as congruence closure of eval (+ folding/unfolding of **letrec**)
  - i.e. equivalence closure over parallel reduction ($Eq\ (\dot{\mathcal{P}} \to\ s\,t)$)

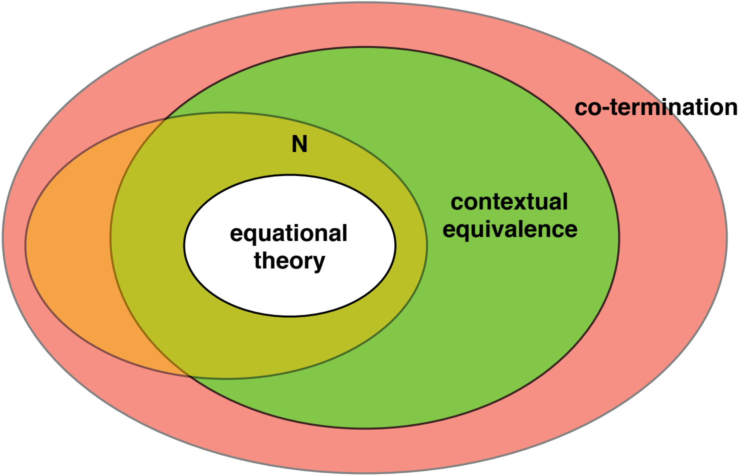
$$\frac{}{\dot{\mathcal{P}}\,R\,x\,x} \qquad \frac{R\,s\,s' \quad \dot{\mathcal{P}}\,R\,s'\,t}{\dot{\mathcal{P}}\,R\,s\,t} \qquad \frac{\dot{\mathcal{P}}\,R\,s\,t}{\dot{\mathcal{P}}\,R\,(\lambda x.\,s)\,(\lambda x.\,t)} \qquad \frac{\dot{\mathcal{P}}\,R\,s\,s' \quad \dot{\mathcal{P}}\,R\,t\,t'}{\dot{\mathcal{P}}\,R\,(s\,t)\,(s'\,t')}$$

- To prove that the equational theory is sound:
  - congruence closure: by definition
  - adequate (i.e. relates terms that co-terminate): using Lassen's normal form bi-simulation

Lassen's Soundness Proof of Equational Theory for CBV



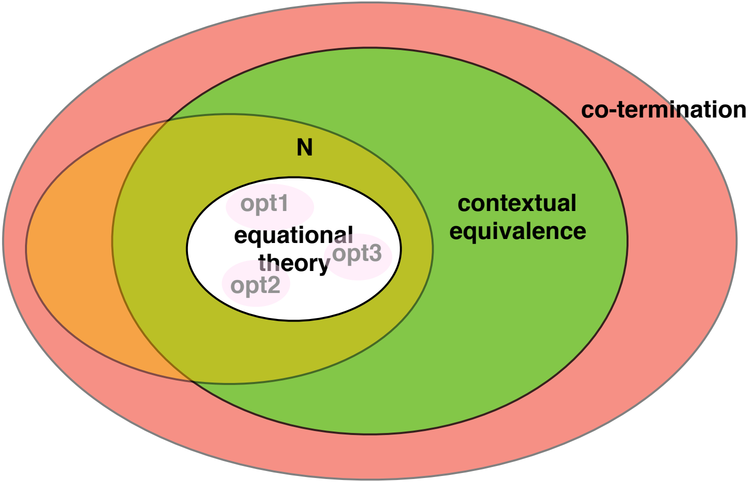
Our Soundness Proof of Equational Theory for CBV and CBPV



Verifying Optimizations

- an optimization is sound if it transforms programs into contextually equivalent programs
- many optimizations are instances of β-reduction, β-expansion, or a step of our CBPV operational semantics in some context
- with our equational theory in place such optimizations are now very easy to verify
- verified several optimizations

Verifying Optimizations using Equational Theory



Examples: Verifying Optimizations

CBPV equation	optimization
$\mathsf{force}\,(\mathsf{thunk}\ M) \equiv M$	block merging "direct jump case"
$V \cdot \lambda x.M \equiv \{V/x\}\,M$	block merging "phi case"
$\mathsf{prd}\ V\ \mathsf{to}\ x\ \mathsf{in}\ M \equiv \{V/x\}\,M$	move elimination
$(n_1 \oplus n_2)\ \mathsf{to}\ x\ \mathsf{in}\ M \equiv \{n_1 [\![\oplus]\!] n_2/x\}\,M$	constant folding
$\mathsf{thunk}\ (\lambda y.M) \cdot \lambda x.N \equiv \{\mathsf{thunk}\ (\lambda y.M)/x\}\,N$	function inlining
$\mathsf{if0}\ 0\ M_1\ M_2 \equiv M_1$	dead branch elimination "true branch"
$\mathsf{if0}\ n\ M_1\ M_2 \equiv M_2$ where $(n \neq 0)$	dead branch elimination "false branch"
$\mathsf{if0}\ V\ M\ M \equiv M$	branch elimination

Call-By-Push-Value

$$\begin{array}{rcll} \text{Values} \ni V & ::= & x \mid n \mid \mathsf{thunk}\, M \\ \text{Terms} \ni M, N & ::= & \mathsf{force}\ V & \mid \ \mathsf{letrec}\, x_1 = M_1, \ldots, x_n = M_n \,\mathsf{in}\, N \\ & \mid & \mathsf{prd}\ V & \mid \ M \,\mathsf{to}\, x \,\mathsf{in}\, N \\ & \mid & V \cdot M & \mid \ \lambda x. M \\ & \mid & V_1 \oplus V_2 & \mid \ \mathsf{if0}\ V\ M_1\ M_2 \\ \text{Sorts} \ni S & ::= & \mathsf{V} \mid \mathsf{C} \end{array}$$

$$\frac{}{M \rightsquigarrow M} \qquad \frac{\{\overline{\mathsf{thunk}\,(\mathsf{letrec}\, \overline{x_i = M_i}^{\,i} \,\mathsf{in}\, M_i)/x_i}^{\,i}\}\, N \rightsquigarrow N'}{\mathsf{letrec}\, \overline{x_i = M_i}^{\,i} \,\mathsf{in}\, N \rightsquigarrow N'}$$

$$\frac{}{\mathsf{force}\,(\mathsf{thunk}\, M) \to M} \qquad \frac{}{\mathsf{if0}\ 0\ M_1\ M_2 \to M_1} \qquad \frac{}{\mathsf{if0}\ n\ M_1\ M_2 \to M_2} \ (n \neq 0)$$

$$\frac{M \rightsquigarrow \mathsf{prd}\ V}{M \,\mathsf{to}\, x \,\mathsf{in}\, N \to \{V/x\}\, N} \qquad \frac{M \rightsquigarrow \lambda x. N}{V \cdot M \to \{V/x\}\, N} \qquad \frac{M \to M'}{V \cdot M \to V \cdot M'}$$

$$\frac{M \to M'}{M \,\mathsf{to}\, x \,\mathsf{in}\, N \to M' \,\mathsf{to}\, x \,\mathsf{in}\, N} \qquad \frac{M \rightsquigarrow (n_1 \oplus n_2)}{M \,\mathsf{to}\, x \,\mathsf{in}\, N \to \{n_1 [\![\oplus]\!] n_2 / x\}\, N}$$

$$\frac{N \rightsquigarrow N' \quad N' \to M}{N \to M} \qquad \frac{\{\overline{\mathsf{thunk}\,(\mathsf{letrec}\, \overline{x_i = M_i}^{\,i} \,\mathsf{in}\, M_i)/x_i}^{\,i}\}\, N \longrightarrow N'}{\mathsf{letrec}\, \overline{x_i = M_i}^{\,i} \,\mathsf{in}\, N \longrightarrow N'}$$

Parallel Reduction (Alternative Definition)

$$\frac{R\, s\, t}{\mathcal{P}\ R\, s\ t} \qquad \frac{\mathcal{P}\ R\, s\ t}{\mathcal{P}\ R\, (\lambda x.\, s)\ (\lambda x.\, t)} \qquad \frac{\mathcal{P}\ R\, s\ s'}{\mathcal{P}\ R\, (s\, t)\ (s'\ t)} \qquad \frac{\mathcal{P}\ R\, t\ t'}{\mathcal{P}\ R\, (s\, t)\ (s\, t')}$$