On the relation between call-by-value and call-by-name

Dylan McDermott   Alan Mycroft

Goal

Suppose we have two semantics for a single language

- e.g. call-by-value and call-by-name

How does replacing one with the other affect the behaviour of programs?

Goal

- Call-by-value: $(\lambda x.\, e)\, e' \rightsquigarrow_{\mathrm{v}}^{*} (\lambda x.\, e)\, v \rightsquigarrow_{\mathrm{v}} e[x \mapsto v] \rightsquigarrow_{\mathrm{v}}^{*} \cdots$
- Call-by-name: $(\lambda x.\, e)\, e' \rightsquigarrow_{\mathrm{n}} e[x \mapsto e'] \rightsquigarrow_{\mathrm{n}}^{*} \cdots$

Goal

- Call-by-value: $(\lambda x.\, e)\, e' \rightsquigarrow_{\mathrm{v}}{}^{*} (\lambda x.\, e)\, v \rightsquigarrow_{\mathrm{v}} e[x \mapsto v] \rightsquigarrow_{\mathrm{v}}{}^{*} \cdots$
- Call-by-name: $(\lambda x.\, e)\, e' \rightsquigarrow_{\mathrm{n}} e[x \mapsto e'] \rightsquigarrow_{\mathrm{n}}{}^{*} \cdots$

If we replace call-by-value with call-by-name, then:

- No side-effects: nothing changes
- Only recursion: behaviour changes, but if CBV terminates with result v, CBN terminates with v

Goal

- Call-by-value: $(\lambda x.\, e)\, e' \rightsquigarrow_{\mathrm{v}}^{*} (\lambda x.\, e)\, v \rightsquigarrow_{\mathrm{v}} e[x \mapsto v] \rightsquigarrow_{\mathrm{v}}^{*} \cdots$
- Call-by-name: $(\lambda x.\, e)\, e' \rightsquigarrow_{\mathrm{n}} e[x \mapsto e'] \rightsquigarrow_{\mathrm{n}}^{*} \cdots$

If we replace call-by-value with call-by-name, then:

- No side-effects: nothing changes
- Only recursion: behaviour changes, but if CBV terminates with result v, CBN terminates with v
- Only nondeterminism: behaviour also different, but if CBV can terminate with result v, then CBN can also terminate with result v
- Mutable state: behaviour changes, we can't say much about how

Questions:

- How can we prove these?
- What properties of the side-effects do we need to prove something?

How to relate different semantics of the same language

1. Define another language that captures both semantics via two sound and adequate translations $(\!|-|\!)^{\mathrm{v}}, (\!|-|\!)^{\mathrm{n}}$

$$\text{(CBV)} \qquad (\!|e|\!)^{\mathrm{v}} \longleftarrow\!\!\!\dashv \; e \; \longmapsto (\!|e|\!)^{\mathrm{n}} \qquad \text{(CBN)}$$

5. For programs (closed, ground expressions) e

$$(\!|e|\!)^{\mathrm{v}} \preccurlyeq (\!|e|\!)^{\mathrm{n}}$$

How to relate different semantics of the same language

1. Define another language that captures both semantics via two sound and adequate translations $(\!|-|\!)^{\mathrm{v}}, (\!|-|\!)^{\mathrm{n}}$

$$\text{(CBV)} \qquad (\!|e|\!)^{\mathrm{v}} \longleftarrow\!\!\!\shortmid \quad e \quad \longmapsto (\!|e|\!)^{\mathrm{n}} \qquad \text{(CBN)}$$

ON THE RELATION BETWEEN DIRECT AND CONTINUATION SEMANTICS†

John C. Reynolds

Systems and Information Science

Syracuse University

5. For programs (closed, ground expressions) e

$$(\!|e|\!)^{\mathrm{v}} \preccurlyeq (\!|e|\!)^{\mathrm{n}}$$

How to relate different semantics of the same language

1. Define another language that captures both semantics via two sound and adequate translations $(\!|-|\!)^{\mathrm{v}}, (\!|-|\!)^{\mathrm{n}}$

$$\text{(CBV)} \qquad (\!|e|\!)^{\mathrm{v}} \;\longleftarrow\!\shortmid\; e \;\longmapsto\; (\!|e|\!)^{\mathrm{n}} \qquad \text{(CBN)}$$

2. Define maps between the two translations

$$\text{CBV translation of } \tau \underset{\Psi_\tau}{\overset{\Phi_\tau}{\rightleftarrows}} \text{CBN translation of } \tau$$

3. Show that Φ, Ψ satisfy nice properties
4. Relate the two translations of (possibly open) expressions e

$$(\!|e|\!)^{\mathrm{v}} \;\preccurlyeq_{\mathrm{ctx}}\; \Psi_\tau((\!|e|\!)^{\mathrm{n}}[\Phi_\Gamma])$$

5. For programs (closed, ground expressions) e

$$(\!|e|\!)^{\mathrm{v}} \;\preccurlyeq\; (\!|e|\!)^{\mathrm{n}}$$

How to relate different semantics of the same language

To relate CBV and CBN:

1. Call-by-push-value [Levy '99] captures CBV and CBN
2. We can define maps Φ_τ, Ψ_τ using the syntax of CBPV
3. Φ and Ψ:
   - behave nicely wrt the CBV and CBN translations, e.g.

     $$\Phi_{\tau_1 \times \tau_2} (\!| e |\!)^{\mathrm{v}} \;=\; (\Phi_{\tau_1} (\!| \mathbf{fst}\, e |\!)^{\mathrm{v}},\; \Phi_{\tau_2} (\!| \mathbf{snd}\, e |\!)^{\mathrm{v}})$$

   - form Galois connections $\Phi_\tau \dashv \Psi_\tau$ (wrt $\preccurlyeq_{\mathrm{ctx}}$) when side-effects are thunkable
4. (3) implies $(\!| e |\!)^{\mathrm{v}} \preccurlyeq_{\mathrm{ctx}} \Psi_\tau((\!| e |\!)^{\mathrm{n}}[\Phi_\Gamma])$
5. (4) is $(\!| e |\!)^{\mathrm{v}} \preccurlyeq (\!| e |\!)^{\mathrm{n}}$ when e is a program

Example

For recursion and nondeterminism, define

$$M_1 \preccurlyeq M_2 \quad \Leftrightarrow \quad \forall V.\ M_1 \Downarrow \mathbf{return}\, V \ \Rightarrow\ M_2 \Downarrow \mathbf{return}\, V$$

($\Downarrow$ is evaluation in CBPV)

so $M_1 \preccurlyeq_{\mathrm{ctx}} M_2$ means

$$\forall V.\ C[M_1] \Downarrow \mathbf{return}\, V \ \Rightarrow\ C[M_2] \Downarrow \mathbf{return}\, V$$

for closed, ground contexts C

Both side-effects are thunkable, so Φ and Φ form Galois connections, so

$$(\!| e |\!)^{\mathrm{v}} \preccurlyeq_{\mathrm{ctx}} \Psi_\tau((\!| e |\!)^{\mathrm{n}}[\Phi_\Gamma])$$

Example

For programs e, we have

$$(\!|e|\!)^{\mathrm{v}} \preccurlyeq (\!|e|\!)^{\mathrm{n}}$$

so

$$\begin{aligned} e \rightsquigarrow_{\mathrm{v}}^{*} v &\iff (\!|e|\!)^{\mathrm{v}} \Downarrow \mathbf{return}\,(\!|v|\!) && \text{(soundness)} \\ &\implies (\!|e|\!)^{\mathrm{n}} \Downarrow \mathbf{return}\,(\!|v|\!) && ((\!|e|\!)^{\mathrm{v}} \preccurlyeq (\!|e|\!)^{\mathrm{n}}) \\ &\iff e \rightsquigarrow_{\mathrm{n}}^{*} v && \text{(adequacy)} \end{aligned}$$

Call-by-push-value [Levy '99]

Split syntax into values and computations

- Values don't reduce, computations do

Call-by-push-value [Levy '99]

Split syntax into values and computations

- Values don't reduce, computations do

Evaluation order is explicit

- Sequencing of computations:

$$\frac{\Gamma \vdash V : A}{\Gamma \mathrel{\underline{\vdash}} \mathbf{return}\, V : \mathbf{F}\, A} \qquad \frac{\Gamma \mathrel{\underline{\vdash}} M_1 : \mathbf{F}\, A \qquad \Gamma, x : A \mathrel{\underline{\vdash}} M_2 : \underline{C}}{\Gamma \mathrel{\underline{\vdash}} M_1 \;\mathbf{to}\; x.\, M_2 : \underline{C}}$$

- Thunks:

$$\frac{\Gamma \mathrel{\underline{\vdash}} M : \underline{C}}{\Gamma \vdash \mathbf{thunk}\, M : \mathbf{U}\, \underline{C}} \qquad \frac{\Gamma \vdash V : \mathbf{U}\, \underline{C}}{\Gamma \mathrel{\underline{\vdash}} \mathbf{force}\, V : \underline{C}}$$

Call-by-value and call-by-name

Source language types:

$$\tau ::= \mathbf{1} \mid \mathbf{2} \mid \tau \to \tau'$$

CBV and CBN translations into CBPV:

$$\begin{array}{rcl} \tau & \mapsto & \text{value type } (\!|\tau|\!)^{\mathrm{v}} \\ \mathbf{1} & \mapsto & \mathbf{1} \\ \mathbf{2} & \mapsto & \mathbf{2} \\ (\tau \to \tau') & \mapsto & \mathbf{U}((\!|\tau|\!)^{\mathrm{v}} \to \mathbf{F}\,(\!|\tau'|\!)^{\mathrm{v}}) \end{array} \qquad \begin{array}{rcl} \tau & \mapsto & \text{computation type } (\!|\tau|\!)^{\mathrm{n}} \\ \mathbf{1} & \mapsto & \mathbf{F}\ \mathbf{1} \\ \mathbf{2} & \mapsto & \mathbf{F}\ \mathbf{2} \\ (\tau \to \tau') & \mapsto & ((\mathbf{U}\,(\!|\tau|\!)^{\mathrm{n}}) \to (\!|\tau'|\!)^{\mathrm{n}}) \end{array}$$

$$\Gamma, x : \tau \ \mapsto\ (\!|\Gamma|\!)^{\mathrm{v}}, x : (\!|\tau|\!)^{\mathrm{v}} \qquad \Gamma, x : \tau \ \mapsto\ (\!|\Gamma|\!)^{\mathrm{n}}, x : \mathbf{U}\,(\!|\tau|\!)^{\mathrm{n}}$$

$$\Gamma \vdash e : \tau \ \mapsto\ (\!|\Gamma|\!)^{\mathrm{v}} \mathrel{\underline{\vdash}} (\!|e|\!)^{\mathrm{v}} : \mathbf{F}\,(\!|\tau|\!)^{\mathrm{v}} \qquad \Gamma \vdash e : \tau \ \mapsto\ (\!|\Gamma|\!)^{\mathrm{n}} \mathrel{\underline{\vdash}} (\!|e|\!)^{\mathrm{n}} : (\!|\tau|\!)^{\mathrm{n}}$$

Call-by-value and call-by-name

Define maps between CBV and CBN:

$$\Gamma \underline{\vdash} M : \mathbf{F}\,(\!|\tau|\!)^{\mathrm{v}} \quad \mapsto \quad \Gamma \underline{\vdash} \Phi_\tau M : \quad (\!|\tau|\!)^{\mathrm{n}} \qquad \text{(CBV to CBN)}$$

$$\Gamma \underline{\vdash} N : \quad (\!|\tau|\!)^{\mathrm{n}} \quad \mapsto \quad \Gamma \underline{\vdash} \Psi_\tau N : \mathbf{F}\,(\!|\tau|\!)^{\mathrm{v}} \qquad \text{(CBN to CBV)}$$

Call-by-value and call-by-name

Define maps between CBV and CBN:

$$\Gamma \vDash M : \mathbf{F}\,(\!|\tau|\!)^{\mathrm{v}} \quad \mapsto \quad \Gamma \vDash \Phi_\tau M : \quad (\!|\tau|\!)^{\mathrm{n}} \qquad \text{(CBV to CBN)}$$

$$\Gamma \vDash N : \quad (\!|\tau|\!)^{\mathrm{n}} \quad \mapsto \quad \Gamma \vDash \Psi_\tau N : \mathbf{F}\,(\!|\tau|\!)^{\mathrm{v}} \qquad \text{(CBN to CBV)}$$

Example: for $\tau = \mathbf{1} \longrightarrow \mathbf{1}$, we have

$$(\!|\mathbf{1} \longrightarrow \mathbf{1}|\!)^{\mathrm{v}} = \mathbf{U}\,(\mathbf{1} \longrightarrow \mathbf{F}\,\mathbf{1})$$
$$(\!|\mathbf{1} \longrightarrow \mathbf{1}|\!)^{\mathrm{n}} = \mathbf{U}\,(\mathbf{F}\,\mathbf{1}) \longrightarrow \mathbf{F}\,\mathbf{1}$$

$$M \quad \overset{\Phi_{\mathbf{1}\to\mathbf{1}}}{\longmapsto} \quad M \ \mathbf{to}\ f.\,\lambda x.\,\mathbf{force}\,x\ \mathbf{to}\ z.\,z\,\text{`}\,\mathbf{force}\,f$$

$$N \quad \overset{\Psi_{\mathbf{1}\to\mathbf{1}}}{\longmapsto} \quad \mathbf{return}\,(\mathbf{thunk}\,(\lambda x.\,(\mathbf{thunk}\,\mathbf{return}\,x)\,\text{`}\,N))$$

Galois connection between CBV and CBN?

Since Φ and Ψ behave nicely wrt translations, e.g.

$$\Phi_{\tau_1 \times \tau_2} (\!| e |\!)^{\mathrm{v}} \;=\; (\Phi_{\tau_1} (\!| \mathbf{fst}\, e |\!)^{\mathrm{v}},\; \Phi_{\tau_2} (\!| \mathbf{snd}\, e |\!)^{\mathrm{v}})$$

if (Φ_τ, Ψ_τ) is a Galois connection (adjunction) for each τ, i.e.

$$M \;\preccurlyeq_{\mathrm{ctx}}\; \Psi_\tau(\Phi_\tau M) \qquad \Phi_\tau(\Psi_\tau N) \;\preccurlyeq_{\mathrm{ctx}}\; N$$

then

$$(\!| e |\!)^{\mathrm{v}} \;\preccurlyeq_{\mathrm{ctx}}\; \Psi_\tau((\!| e |\!)^{\mathrm{n}}[\Phi_\Gamma])$$

Galois connection between CBV and CBN?

These do not always hold!

$$M \;\preccurlyeq_{\text{ctx}}\; \Psi_\tau(\Phi_\tau M) \qquad \Phi_\tau(\Psi_\tau N) \;\preccurlyeq_{\text{ctx}}\; N$$

- Don't hold for: exceptions, mutable state

$$\mathbf{raise} \not\preccurlyeq_{\text{ctx}} \mathbf{return}\,(\dots) = \Psi_{\mathbf{1}\to\mathbf{1}}(\Phi_{\mathbf{1}\to\mathbf{1}}\,\mathbf{raise})$$
$$\big(\diamond \vdash \mathbf{raise} : \mathbf{F}\,(\mathbf{U}\,(\mathbf{1} \to \mathbf{F}\,\mathbf{1}))\big)$$

- Do hold for: no side-effects, recursion, nondeterminism

This is where the side-effects matter

Galois connection between CBV and CBN?

Definition (Thunkable [Führmann '99])

A computation $\Gamma \vDash M : \mathbf{F}\,A$ is (lax) *thunkable* if

$$M\ \mathbf{to}\ x.\,\mathbf{return}\,(\mathbf{thunk}\,(\mathbf{return}\,x)) \ \ \preccurlyeq_{\mathrm{ctx}}\ \ \mathbf{return}\,(\mathbf{thunk}\,M)$$

- Essentially: we're allowed to suspend the computation M
- Implies M commutes with other computations, is (lax) discardable, (lax) copyable

Galois connection between CBV and CBN?

Definition (Thunkable [Führmann '99])

A computation $\Gamma \vDash M : \mathbf{F}\,A$ is (lax) *thunkable* if

$$M\ \mathbf{to}\ x.\,\mathbf{return}\,(\mathbf{thunk}\,(\mathbf{return}\,x)) \ \preccurlyeq_{\mathrm{ctx}}\ \mathbf{return}\,(\mathbf{thunk}\,M)$$

- Essentially: we're allowed to suspend the computation M
- Implies M commutes with other computations, is (lax) discardable, (lax) copyable

Lemma

If every computation is thunkable, then (Φ_τ, Ψ_τ) is a Galois connection.

How to relate call-by-value to call-by-name

If every computation is thunkable then

$$(\!|e|\!)^{\mathrm{v}} \;\; \preccurlyeq_{\mathrm{ctx}} \;\; \Psi_\tau((\!|e|\!)^{\mathrm{n}}[\Phi_\Gamma])$$

for each e. (And the converse holds for computations of ground type.)

And if e is a program then

$$(\!|e|\!)^{\mathrm{v}} \preccurlyeq (\!|e|\!)^{\mathrm{n}}$$

Examples

If e is a program:

- No side-effects: $(\!|e|\!)^{\mathrm{v}}$ and $(\!|e|\!)^{\mathrm{n}}$ reduce to the same values
- Nontermination: if $(\!|e|\!)^{\mathrm{v}}$ reduces to v, then so does $(\!|e|\!)^{\mathrm{n}}$
- Nondeterminism: if $(\!|e|\!)^{\mathrm{v}}$ can reduce to v, then $(\!|e|\!)^{\mathrm{n}}$ can also reduce to v

But this doesn't prove anything about exceptions, state, ...

Overview

How to relate evaluation orders:

1. Translate from source language to intermediate language
2. Define maps between evaluation orders
3. Relate terms:

$$(\!|e|\!)^{\mathrm{v}} \;\; \preccurlyeq_{\mathrm{ctx}} \;\; \Psi_{\tau}((\!|e|\!)^{\mathrm{n}}[\Phi_{\Gamma}])$$

- Works for call-by-value and call-by-name
- Also works for other things like comparing direct and continuation-style semantics [Reynolds '74], strict and lazy products, etc.