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: \((\lambda x. e) e' \Rightarrow_v^* (\lambda x. e) v \Rightarrow_v e[x \mapsto v] \Rightarrow_v^* \cdots\)
- Call-by-name: \((\lambda x. e) e' \Rightarrow_n e[x \mapsto e'] \Rightarrow_n^* \cdots\)

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?
Goal

▶ Call-by-value: \((\lambda x. e) e' \rightsquigarrow_v^* (\lambda x. e) v \rightsquigarrow_v e[x \mapsto v] \rightsquigarrow_v^* \cdots\)
▶ Call-by-name: \((\lambda x. e) e' \rightsquigarrow_n e[x \mapsto e'] \rightsquigarrow_n^* \cdots\)

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

▶ No side-effects: nothing changes
▶ Only recursion: behaviour changes

\[
\text{CBV: } (\lambda x. \text{false}) \Omega \rightsquigarrow_v (\lambda x. \text{false}) \Omega \rightsquigarrow_v \cdots
\]
\[
\text{CBN: } (\lambda x. \text{false}) \Omega \rightsquigarrow_n \text{false}
\]

but if CBV terminates with result \(v\), CBN terminates with \(v\)
Goal

- Call-by-value: $((\lambda x. e) \ e') \rightarrow^* (\lambda x. e) \ v \rightarrow v \ e[x \mapsto v] \rightarrow^* \cdots$
- Call-by-name: $((\lambda x. e) \ e') \rightarrow_n e[x \mapsto e'] \rightarrow^* \cdots$

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 $\langle - \rangle^v, \langle - \rangle^n$

   \[
   (\text{CBV}) \quad \langle e \rangle^v \longleftrightarrow e \quad \longleftrightarrow \quad \langle e \rangle^n \quad (\text{CBN})
   \]

5. For programs (closed, ground expressions) $e$

   \[
   \langle e \rangle^v \preceq \langle e \rangle^n
   \]
How to relate different semantics of the same language

1. Define another language that captures both semantics via two sound and adequate translations \( (\vdash) \lor, (\vdash) \land \)

\[
\begin{align*}
\text{(CBV)} & \quad (e) \lor \leftrightarrow e \leftrightarrow (e) \land \\
\text{\text{(CBN)} } & \quad \text{(CBN)}
\end{align*}
\]

ON THE RELATION BETWEEN DIRECT AND CONTINUATION SEMANTICS

John C. Reynolds
Systems and Information Science
Syracuse University

5. For programs (closed, ground expressions) \( e \)

\[
(e) \lor \preceq (e) \land
\]
How to relate different semantics of the same language

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

   (CBV) \( (|e|^v \leftrightarrow e \longrightarrow (|e|^n) \) (CBN)

2. Define maps between the two translations

   CBV translation of \( \tau \) \( \Phi_\tau \) CBN translation of \( \tau \) \( \Psi_\tau \)

3. Show that \( \Phi, \Psi \) satisfy nice properties

4. Relate the two translations of (possibly open) expressions \( e \)

   \( (|e|^v \leq_{ctx} \Psi_\tau((|e|^n[\Phi_\Gamma])) \)

5. For programs (closed, ground expressions) \( e \)

   \( (|e|^v \leq (|e|^n) \)
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 $\Phi_\tau, \Psi_\tau$ using the syntax of CBPV
3. $\Phi$ and $\Psi$:
   - behave nicely wrt the CBV and CBN translations, e.g.
     $$\Phi_{\tau_1 \times \tau_2}(|e|^v) = (\Phi_{\tau_1}(|\text{fst } e|^v), \Phi_{\tau_2}(|\text{snd } e|^v))$$
   - form Galois connections $\Phi_\tau \dashv \Psi_\tau$ (wrt $\leq_{\text{ctx}}$) when side-effects are thunkable
4. (3) implies $|e|^v \leq_{\text{ctx}} \Psi_\tau(|e|^n[\Phi_\Gamma])$
5. (4) is $|e|^v \leq |e|^n$ when $e$ is a program
Example

For recursion and nondeterminism, define

\[ M_1 \preceq M_2 \iff \forall V. M_1 \downarrow \text{return } V \Rightarrow M_2 \downarrow \text{return } V \]

(\(\downarrow\) is evaluation in CBPV)

so \(M_1 \preceq_{\text{ctx}} M_2\) means

\[ \forall V. C[M_1] \downarrow \text{return } V \Rightarrow C[M_2] \downarrow \text{return } V \]

for closed, ground contexts \(C\)

Both side-effects are thunkable, so \(\Phi\) and \(\Phi\) form Galois connections, so

\[ (e)^v \preceq_{\text{ctx}} \Psi_\tau((e)^n[\Phi_\Gamma]) \]
Example

For programs $e$, we have

$$(|e|^v) \leq (|e|^n)$$

so

$$e \leadsto^*_v v \iff (|e|^v \downarrow \text{return} (|v|)$$

$$\Rightarrow (|e|^n \downarrow \text{return} (|v|)$$

$$(|e|^v) \leq (|e|^n)$$

$$\iff e \leadsto^*_n v$$

(soundness)

(adequacy)
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:

\[
\Gamma \vdash V : A \\
\Gamma \vdash return V : FA\\n\Gamma \vdash M_1 : FA \\
\Gamma, x : A \vdash M_2 : C\\n\Gamma \vdash M_1 \text{ to } x. M_2 : C
\]

- Thunks:

\[
\Gamma \vdash M : C \\
\Gamma \vdash thunk M : UC \\
\Gamma \vdash V : UC \\
\Gamma \vdash force V : C
\]
Call-by-value and call-by-name

Source language types:

\[ \tau ::= 1 | 2 | \tau \rightarrow \tau' \]

CBV and CBN translations into CBPV:

\[ \begin{align*}
\tau & \mapsto \text{value type } (|\tau|)^V \\
1 & \mapsto 1 \\
2 & \mapsto 2 \\
(\tau \rightarrow \tau') & \mapsto U(|\tau|)^V \rightarrow F(|\tau'|)^V \\
\end{align*} \]

\[ \begin{align*}
\tau & \mapsto \text{computation type } (|\tau|)^n \\
1 & \mapsto F \, 1 \\
2 & \mapsto F \, 2 \\
(\tau \rightarrow \tau') & \mapsto (U(|\tau|)^n) \rightarrow (|\tau'|)^n \\
\end{align*} \]

\[ \begin{align*}
\Gamma, x : \tau & \mapsto (|\Gamma|)^V, x : (|\tau|)^V \\
\Gamma, x : \tau & \mapsto (|\Gamma|)^n, x : U(|\tau|)^n \\
\Gamma \vdash e : \tau & \mapsto (|\Gamma|)^V \vdash (|e|)^V : F(|\tau|)^V \\
\Gamma \vdash e : \tau & \mapsto (|\Gamma|)^n \vdash (|e|)^n : (|\tau|)^n \\
\end{align*} \]
Call-by-value and call-by-name

Define maps between CBV and CBN:

\[ \Gamma \vdash M : \text{F}(\tau)^V \quad \mapsto \quad \Gamma \vdash \Phi_\tau M : \text{(\tau)}^n \quad \text{(CBV to CBN)} \]

\[ \Gamma \vdash N : \text{(\tau)}^n \quad \mapsto \quad \Gamma \vdash \Psi_\tau N : \text{F}(\tau)^V \quad \text{(CBN to CBV)} \]
Call-by-value and call-by-name

Define maps between CBV and CBN:

\[ \Gamma \vdash M : F\,|\tau|^{V} \quad \mapsto \quad \Gamma \vdash \Phi_{\tau}M : (|\tau|)^{n} \quad \text{(CBV to CBN)} \]

\[ \Gamma \vdash N : (|\tau|)^{n} \quad \mapsto \quad \Gamma \vdash \Psi_{\tau}N : F\,|\tau|^{V} \quad \text{(CBN to CBV)} \]

Example: for \( \tau = 1 \rightarrow 1 \), we have

\[ (|1 \rightarrow 1|^{V} = U\,(1 \rightarrow F\,1) \]

\[ (|1 \rightarrow 1|^{n} = U\,(F\,1) \rightarrow F\,1 \]

\[ M \quad \Phi_{1\rightarrow 1} \quad M \text{ to } f.\,\lambda x.\,\text{force } x \text{ to } z.\,z' \text{ force } f \]

\[ N \quad \Psi_{1\rightarrow 1} \quad \text{return } (\text{thunk } (\lambda x.\,(\text{thunk return } x) \,\text{'} \,N)) \]
Galois connection between CBV and CBN?

Since $\Phi$ and $\Psi$ behave nicely wrt translations, e.g.

$$\Phi_{\tau_1 \times \tau_2}(|e|)^v = (\Phi_{\tau_1}(|\text{fst } e|)^v, \Phi_{\tau_2}(|\text{snd } e|)^v)$$

if $(\Phi_\tau, \Psi_\tau)$ is a Galois connection (adjunction) for each $\tau$, i.e.

$$M \leq_{\text{ctx}} \Psi_\tau(\Phi_\tau M) \quad \Phi_\tau(\Psi_\tau N) \leq_{\text{ctx}} N$$

then

$$|e|^{v} \leq_{\text{ctx}} \Psi_\tau(|e|^{n}[\Phi_\Gamma])$$
Galois connection between CBV and CBN?

These do not always hold!

\[ M \preceq_{\text{ctx}} \Psi_{\tau}(\Phi_{\tau}M) \quad \Phi_{\tau}(\Psi_{\tau}N) \preceq_{\text{ctx}} N \]

- Don’t hold for: exceptions, mutable state

\[ \text{raise} \not\preceq_{\text{ctx}} \text{return} (\ldots) = \Psi_{1\rightarrow1}(\Phi_{1\rightarrow1} \text{raise}) \]
\[ (\diamond \vdash \text{raise} : F(U(1 \rightarrow F1))) \]

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

This is where the side-effects matter
Definition (Thunkable [Führmann '99])
A computation \( \Gamma \vdash M : FA \) is (lax) thunkable if

\[
M \text{ to } x. \text{return} (\text{thunk} (\text{return} x)) \preceq_{\text{ctx}} \text{return} (\text{thunk} M)
\]

- 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 $\Gamma \vdash M : FA$ is (lax) thunkable if

$$M \text{ to } x. \text{return} (\text{thunk} (\text{return} x)) \preceq_{\text{ctx}} \text{return} (\text{thunk} M)$$

- 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 $(\Phi_\tau, \Psi_\tau)$ is a Galois connection.
How to relate call-by-value to call-by-name

If every computation is thunkable then

\[(|e|)^v \leq_{ctx} \Psi_{\tau}((|e|^n[\Phi\Gamma]))\]

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

And if \(e\) is a program then

\[(|e|)^v \leq (|e|^n)\]
Overview

How to relate two different semantics:

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

   \[(\|e\|)^v \leq_{ctx} \Psi_\tau(\|e\|^n[\Phi_\Gamma])\]

   - 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.