Computational Soundness of a Call by Name Calculus of Recursively-scoped Records

Elena Machkasova

University of Minnesota, Morris

WRS 2007

Outline

Overview of the calculus

- Untyped CBN calculus
- Records are unordered collections of labeled terms
- Records represent mutual dependencies, including cyclic dependencies
- Cyclic dependencies arise in separate compilation, modules and linking, `letrec`.

Overview of records

Example of a record:

$[l_1 \mapsto 2+3,\;\; l_2 \mapsto \lambda x.x,\;\; l_3 \mapsto l_2 \;@\; l_1]$

- 3 components, with labels l_1, l_2, l_3
- labels are bound to λ-terms
- components reference each other via labels

Evaluation $\Rightarrow$ of a record (leftmost, outermost strategy):

$$
\begin{array}{ll}
[l_1 \mapsto 2+3,\;\; l_2 \mapsto \lambda x.x,\;\; l_3 \mapsto l_2 \;@\; l_1] & \Rightarrow \\
[l_1 \mapsto 2+3,\;\; l_2 \mapsto \lambda x.x,\;\; l_3 \mapsto (\lambda x.x) \;@\; l_1] & \Rightarrow \\
[l_1 \mapsto 2+3,\;\; l_2 \mapsto \lambda x.x,\;\; l_3 \mapsto l_1] & \Rightarrow \\
[l_1 \mapsto 2+3,\;\; l_2 \mapsto \lambda x.x,\;\; l_3 \mapsto 2+3] & \Rightarrow \\
[l_1 \mapsto 5,\;\; l_2 \mapsto \lambda x.x,\;\; l_3 \mapsto 2+3] & \Rightarrow \ldots
\end{array}
$$

At most one evaluation step is possible in each component.

Overview of records (cont.)

A rewriting relation $\rightarrow$:

$$
\begin{array}{ll}
[l_1 \mapsto 2+3,\ \ l_2 \mapsto \lambda x.x,\ \ l_3 \mapsto l_2 \mathbin{@} l_1] & \rightarrow \\
[l_1 \mapsto 2+3,\ \ l_2 \mapsto \lambda x.x,\ \ l_3 \mapsto l_2 \mathbin{@} (2+3)] & \rightarrow \\
[l_1 \mapsto 2+3,\ \ l_2 \mapsto \lambda x.x,\ \ l_3 \mapsto l_2 \mathbin{@} 5] & \rightarrow \\
[l_1 \mapsto 2+3,\ \ l_2 \mapsto \lambda x.x,\ \ l_3 \mapsto (\lambda x.x) \mathbin{@} 5] & \rightarrow
\end{array}
$$

Computational soundness: rewriting steps preserve the meaning of a term, as defined by $\Rightarrow$.

Term-level calculus

Terms and term contexts:

$$
\begin{array}{lcl}
M & ::= & c \mid x \mid l \mid \bullet \mid \lambda x.M \mid M_1 \mathbin{@} M_2 \mid M_1 + M_2 \\
\mathbb{C} & ::= & \Box \mid \lambda x.\mathbb{C} \mid \mathbb{C} \mathbin{@} M \mid M \mathbin{@} \mathbb{C} \mid \mathbb{C} + M \mid M + \mathbb{C} \\
\mathbb{E} & ::= & \Box \mid \mathbb{E} \mathbin{@} M \mid \mathbb{E} + M \mid c + \mathbb{E}
\end{array}
$$

c - constants, x, y, z - variables, l - labels, $\bullet$ - black hole.
$\mathbb{C}$ - general context (the hole may be anywhere in a term), $\mathbb{E}$ - evaluation context.
$\mathbb{C}\{M\}$ is the result of $\mathbb{C}$ with M.
Terms: $\lambda x.2 + 3$, $(\lambda x.x) \mathbin{@} \bullet$, $l_1 + 2$
Evaluation contexts: $\Box$, $\Box + l_1$, $\Box \mathbin{@} \lambda x.x$
Non-evaluation general contexts: $\lambda x.\Box$, $(\lambda x.x) \mathbin{@} \Box$

Relations on terms

$\rightsquigarrow$ - the elementary reduction, $\Rightarrow$ - evaluation, $\rightarrow$ - rewriting relation (reduction).

$$\begin{array}{lcll}
(\lambda x.M) \mathbin{@} N & \rightsquigarrow & M[x := N] & (\beta) \\
c_1 \; + \; c_2 & \rightsquigarrow & c_3 \text{ (the result of the operation } +) & (\text{op}) \\
\mathbb{C}\{R\} & \rightarrow & \mathbb{C}\{Q\} \text{ where } R \rightsquigarrow Q & \\
\mathbb{E}\{R\} & \Rightarrow & \mathbb{E}\{Q\} \text{ where } R \rightsquigarrow Q &
\end{array}$$

Non-evaluation: $\hookrightarrow \; = \; \rightarrow \; \setminus \; \Rightarrow$

Examples:

$$(\lambda x.x) \mathbin{@} (2+3) \Rightarrow 2+3$$
$$(\lambda x.x) \mathbin{@} (2+3) \hookrightarrow (\lambda x.x) \mathbin{@} 5$$

Record calculus

Records:

$$\begin{array}{lcl} D & ::= & [l_1 \mapsto M_1, ..., l_n \mapsto M_n],\ l_i \neq l_j \text{ for } i \neq j \\ \mathbb{D} & ::= & [l \mapsto \mathbb{C}, l_1 \mapsto M_1, \ldots, l_n \mapsto M_n] \text{ record context} \\ \mathbb{G} & ::= & [l \mapsto \mathbb{E}, l_1 \mapsto M_1, \ldots, l_n \mapsto M_n] \text{ record eval. context,} \end{array}$$

$\mathbb{C}$ is a term context, $\mathbb{E}$ is a term evaluation context.

Records: $[l_1 \mapsto 2 + 3, l_2 \mapsto \lambda x.x, l_3 \mapsto l_2 \mathbin{@} l_1]$,
$[l_1 \mapsto \bullet, l_2 \mapsto \lambda x.l_1]$

Evaluation context: $[l_1 \mapsto \square + 2, l_2 \mapsto \lambda x.x, l_3 \mapsto l_2 \mathbin{@} l_1]$

Non-evaluation context: $[l_1 \mapsto 2 + 3, l_2 \mapsto \lambda x.\square, l_3 \mapsto l_2 \mathbin{@} l_1]$

Relations on records: term reduction

Term reduction: reducing a component in a record.

$$\begin{array}{lcll} \mathbb{D}\{R\} & \rightarrow & \mathbb{D}\{Q\}, R \rightsquigarrow Q & (T) \\ \mathbb{G}\{R\} & \Rightarrow & \mathbb{G}\{Q\}, R \rightsquigarrow Q & (TE) \end{array}$$

Examples:

$$\begin{array}{ll} [l_1 \mapsto 2+3, l_2 \mapsto \lambda x.x, l_3 \mapsto l_2 \mathbin{@} l_1] & \Rightarrow \\ [l_1 \mapsto 5, l_2 \mapsto \lambda x.x, l_3 \mapsto l_2 \mathbin{@} l_1] & \\ [l_1 \mapsto \lambda x.2+3, l_2 \mapsto \lambda x.x, l_3 \mapsto l_2 \mathbin{@} l_1] & \hookrightarrow \\ l_1 \mapsto \lambda x.5, l_2 \mapsto \lambda x.x, l_3 \mapsto l_2 \mathbin{@} l_1] & \end{array}$$

Relations on records: substitution

Substitution:

$$\begin{array}{lcll} \mathbb{D}\{l\} & \rightarrow & \mathbb{D}\{M\},\ l \mapsto M \in \mathbb{D}\{l\},\ \mathbb{D} \neq [l \mapsto \mathbb{E}, \ldots] & (S) \\ \mathbb{G}\{l\} & \Rightarrow & \mathbb{G}\{M\},\ l \mapsto M \in \mathbb{G}\{l\},\ \mathbb{G} \neq [l \mapsto \mathbb{E}, \ldots] & (SE) \end{array}$$

Examples:

$$\begin{array}{ll} [l_1 \mapsto 2+3, l_2 \mapsto \lambda x.x, l_3 \mapsto l_2 \mathbin{@} l_1] & \Rightarrow \\ [l_1 \mapsto 2+3, l_2 \mapsto \lambda x.x, l_3 \mapsto (\lambda x.x) \mathbin{@} l_1] & \\ [l_1 \mapsto 2+3, l_2 \mapsto \lambda x.x, l_3 \mapsto l_2 \mathbin{@} l_1] & \hookrightarrow \\ [l_1 \mapsto 2+3, l_2 \mapsto \lambda x.x, l_3 \mapsto l_2 \mathbin{@} (2+3)] & \end{array}$$

Relations on records: black hole

Black hole $\bullet$ denotes apparent infinite substitution cycles.
Black hole reductions:

$$\begin{array}{lcll} [l_1 \mapsto \mathbb{E}\{l_1\}, ...] & \Rightarrow & [l_1 \mapsto \bullet, ...] & (B1) \\ [l_1 \mapsto \mathbb{E}\{\bullet\}, ...] & \Rightarrow & [l_1 \mapsto \bullet, ...] & (B2) \end{array}$$

(B1) – introduction of $\bullet$:

$$[l_1 \mapsto l_1 + 1] \Rightarrow [l_1 \mapsto \bullet]$$

(instead of $[l_1 \mapsto l_1 + 1] \Rightarrow [l_1 \mapsto l_1 + 1 + 1] \Rightarrow \ldots$)
(B2) – propagation of $\bullet$:

$$[l_1 \mapsto \bullet, l_2 \mapsto l_1 + 1] \Rightarrow [l_1 \mapsto \bullet, l_2 \mapsto \bullet + 1] \Rightarrow [l_1 \mapsto \bullet, l_2 \mapsto \bullet]$$

Confluence of evaluation

Lemma (Confluence of Evaluation)

$\Rightarrow$ is confluent on records.

A potential non-confluence example (similar to one in Ariola, Klop 1996):

$$[l_1 \mapsto 2 + l_2, l_2 \mapsto l_1 + 1] \Rightarrow [l_1 \mapsto 2 + l_1 + 1, l_2 \mapsto l_1 + 1]$$
$$[l_1 \mapsto 2 + l_2, l_2 \mapsto l_1 + 1] \Rightarrow [l_1 \mapsto 2 + l_2, l_2 \mapsto 2 + l_2 + 1]$$

Without a black hole both components in one record reference l_1, both components in the second record reference l_2.
With a black hole both records evaluate to $[l_1 \mapsto \bullet, l_2 \mapsto \bullet]$:

$$[l_1 \mapsto 2 + l_1 + 1, l_2 \mapsto l_1 + 1] \Rightarrow [l_1 \mapsto \bullet, l_2 \mapsto l_1 + 1] \Rightarrow$$
$$[l_1 \mapsto \bullet, l_2 \mapsto \bullet + 1] \Rightarrow [l_1 \mapsto \bullet, l_2 \mapsto \bullet]$$

Uniform normalization of $\Rightarrow$

Lemma

Given a record D, if there exists D′ s.t.

- $D \Longrightarrow^* D'$
- D' *is a normal form w.r.t.* $\Rightarrow$,
- *no component in* D' *is bound to* $\bullet$,

then there is no infinite sequence $D \Rightarrow D_1 \Rightarrow D_2 \ldots$.

Classification of terms

Terms are grouped into classes denoted by symbols, possibly with parameters. Terms in the same class have the same "meaning". $Cl(M)$ denotes the class of M:

- $Cl(\mathbb{E}\{R\}) = \textbf{eval}$ if R is a redex. Such terms are called *evaluatable*.
- $Cl(c) = \textbf{const}(c)$, where $\textbf{const}(c_1) = \textbf{const}(c_2)$ if and only if $c_1 = c_2$. i.e. $\textbf{const}(2) \neq \textbf{const}(3)$
- $Cl(\bullet) = \bullet$
- $Cl(\lambda x.N) = \textbf{abs}$
- $Cl(\mathbb{E}\{l\}) = \textbf{stuck}(l)$, where $\textbf{stuck}(l_1) = \textbf{stuck}(l_2)$ if and only if $l_1 = l_2$
- $Cl(M) = \textbf{error}$ otherwise

Classification of records

A class of a record is determined by classes of its components:

- $Cl([l_1 \mapsto M_1, \ldots l_n \mapsto M_n]) = [l_1 \mapsto Cl(M_1), \ldots l_n \mapsto Cl(M_n)]$ if $Cl(M_i) \neq \bullet$ for all i s.t. $1 \leq i \leq n$
- $Cl([l \mapsto \bullet, \ldots]) = \bot$

Example:

$$Cl([l_1 \mapsto \lambda x.x, l_2 \mapsto l_1 \mathbin{@} 1]) = [l_1 \mapsto \mathbf{abs}, l_2 \mapsto \mathbf{stuck}(l_1)]$$

A black hole in an evaluation context represents infinite divergence:

$$Cl([l_1 \mapsto \bullet, l_2 \mapsto 2 + 3]) = \bot$$

Outcome and computational soundness

The *outcome* of a record D, denoted $\mathrm{Outcome}(D)$, is:

- $Cl(D')$ where D' is the normal form of D w.r.t. $\Rightarrow$ if D has a normal form
- $\perp$ if evaluation of D diverges.

A relation R is *meaning preserving* if MRN implies that $\mathrm{Outcome}(M) = \mathrm{Outcome}(N)$.
A calculus is *computationally sound* if the reflexive, symmetric, transitive closure of $\rightarrow$ is meaning preserving.

Theorem

Calculus of records is computationally sound.

$\Rightarrow$ is meaning-preserving by confluence and uniform normalization. Need to prove that $\hookrightarrow$ is meaning-preserving.

Black hole and computational soundness

Some challenges in proving computational soundness:

$$[l_1 \mapsto l_2 @ 2, l_2 \mapsto \lambda x.l_1] \hookrightarrow [l_1 \mapsto l_2 @ 2, l_2 \mapsto \lambda x.l_2 @ 2]$$

The first record evaluates to a n.f. with a black hole:

$$\begin{array}{ll}
[l_1 \mapsto l_2 @ 2, l_2 \mapsto \lambda x.l_1] & \Rightarrow \\
[l_1 \mapsto (\lambda x.l_1) @ 2, l_2 \mapsto \lambda x.l_1] & \Rightarrow \\
[l_1 \mapsto l_1, l_2 \mapsto \lambda x.l_1] & \Rightarrow \ldots \\
[l_1 \mapsto \bullet, l_2 \mapsto \lambda x.\bullet] &
\end{array}$$

The second one diverges:

$$\begin{array}{ll}
[l_1 \mapsto l_2 @ 2, l_2 \mapsto \lambda x.l_2 @ 2] & \Rightarrow \\
[l_1 \mapsto (\lambda x.l_2 @ 2) @ 2, l_2 \mapsto \lambda x.l_2 @ 2] & \Rightarrow \\
[l_1 \mapsto l_2 @ 2, l_2 \mapsto \lambda x.l_2 @ 2] & \Rightarrow \quad \ldots
\end{array}$$

Meaning preservation of term reduction

Meaning preservation of a term reduction is proven using the lift/project approach (introduced in Machkasova&Turbak, 2000). *Lift* and *project* diagrams:

$$
\begin{tikzcd}
D_1 \arrow[r, Rightarrow, dashed, "*"] \arrow[d, hook, "T"'] & D_4 \arrow[d, hook, dashed, "*"] \\
D_2 \arrow[r, Rightarrow, "*"] & D_3
\end{tikzcd}
\qquad
\begin{tikzcd}
D_1 \arrow[r, Rightarrow, "*"] \arrow[d, hook, "T"'] & D_2 \arrow[r, Rightarrow, dashed, "*"] & D_4 \arrow[d, hook, dashed, "*"] \\
D_3 \arrow[rr, Rightarrow, dashed, "*"] & & D_5
\end{tikzcd}
$$

Class preservation: if $D_1 \hookrightarrow D_2$ then $Cl(D_1) = Cl(D_2)$.
If D_3 in *lift* is a normal form w.r.t. $\Rightarrow$, we obtain equivalence of outcomes of D_1 and D_2. Similarly assuming that D_2 in *project* is a normal form.

Efficient evaluation strategy

Efficient evaluation strategy: a partial order on evaluation of record components; similar to *call-by-need*.
Let $D = [l \mapsto M, \ldots]$. The efficient strategy to evaluate l is defined as:

- If $M = \mathbb{E}\{R\}$, evaluate R.
- If $M = \mathbb{E}\{l'\}$ and l' is evaluated to M', substitute M' for l'.
- If $M = \mathbb{E}\{l'\}$ and M' is not a normal form, start evaluating M' using the efficient strategy.
- If M depends on $\bullet$ or on l directly or transitively, then the efficient strategy stops and reports a cycle.
- If M is a substitution-free normal form, the efficient strategy for l in D is undefined.

Efficient evaluation strategy: example

A sequence that follows the efficient strategy; l_1 is the target label:

$$
\begin{array}{ll}
[l_1 \mapsto l_2, l_2 \mapsto l_3 + 2, l_3 \mapsto 1 + 3] & \Rightarrow \\
[l_1 \mapsto l_2, l_2 \mapsto l_3 + 2, l_3 \mapsto 4] & \Rightarrow \\
[l_1 \mapsto l_2, l_2 \mapsto 4 + 2, l_3 \mapsto 4] & \Rightarrow \\
[l_1 \mapsto l_2, l_2 \mapsto 6, l_3 \mapsto 4] & \Rightarrow \\
[l_1 \mapsto 6, l_2 \mapsto 6, l_3 \mapsto 4] &
\end{array}
$$

A valid evaluation, but not efficient strategy (duplicated a redex):

$$
\begin{array}{ll}
[l_1 \mapsto l_2, l_2 \mapsto l_3 + 2, l_3 \mapsto 1 + 3] & \Rightarrow \\
[l_1 \mapsto l_3 + 2, l_2 \mapsto l_3 + 2, l_3 \mapsto 1 + 3] & \Rightarrow \ldots
\end{array}
$$

Any evaluation normal form can be reached by an efficient strategy.

(M_1, M_2)-similarity

Multihole contexts:

$$\mathbb{M} \quad ::= \quad \square \mid M \mid \lambda x.\mathbb{M} \mid \mathbb{M} + \mathbb{M} \mid \mathbb{M} \mathbin{@} \mathbb{M}$$

A record D_1 is called (M_1, M_2)*-similar* to a record D_2 (denoted $D_1 \sim^{M_1}_{M_2} D_2$) if there exist multi-hole contexts $\mathbb{M}_1, \ldots, \mathbb{M}_n$ s.t.

$$\begin{aligned} D_1 &= [l_1 \mapsto \mathbb{M}_1\{M_1, \ldots, M_1\}, \ldots, l_n \mapsto \mathbb{M}_n\{M_1, \ldots, M_1\}], \\ D_2 &= [l_1 \mapsto \mathbb{M}_1\{M_2, \ldots, M_2\}, \ldots, l_n \mapsto \mathbb{M}_n\{M_2, \ldots, M_2\}]. \end{aligned}$$

This means that some occurrences of M_1 in D_1 are replaced by M_2 in D_2.

Meaning preservation of substitution

Suppose $D_1 \hookrightarrow D_2$ by a substiting a term M bound to l into a component labeled l'. Then $D_1 \sim^l_M D_2$ (base case).
Use efficient evaluation strategy starting with labels l, l' (induction on the number of $\Rightarrow$ steps).
We prove that D_1 reaches a black-hole-free normal form if and only if D_2 does, and the resulting records remain (l, M)-similar:

$$
\begin{tikzcd}
D_1 \arrow[r, Rightarrow, "*"] \arrow[d, hook] & D_1' \arrow[d, squiggly, no head, "\sim^l_M"] \\
D_2 \arrow[r, Rightarrow, dashed, "*"] & D_2'
\end{tikzcd}
\qquad
\begin{tikzcd}
D_1 \arrow[rr, Rightarrow, dashed, "*"] \arrow[d, hook] & & D_1' \arrow[d, squiggly, no head, "\sim^l_M"] \\
D_2 \arrow[r, Rightarrow, "*"] & D_2' \arrow[r, Rightarrow, dashed, "*"] & D_2''
\end{tikzcd}
$$

If both records evaluate to normal forms then the differences are only in non-evaluation contexts, don't effect the class of n.f. (i.e the outcome).

Conclusions

- We have proven that a CBN system of mutually recursive components is computationally sound.
- Diagram-based approaches are problematic for such systems.
- The context-based method allows us to prove computational soundness.

Future work:

- Study applicability of the context method to other systems with cyclic dependencies: `letrec` calculi; modules and linking
- Continue comparison with other methods of proving computational soundness.

Selected bibliography

More detailed presentation, see http://cda.morris.umn.edu/~elenam/

- E. Machkasova: Computational Soundness of a Call by Name Calculus of Recursively-scoped Records. Working Papers Series, University of Minnesota, Morris, Vol. 2 Num. 3, 2007.
- E. Machkasova, E. Christiansen: Call-by-name Calculus of Records and its Basic Properties. Working Papers Series, University of Minnesota, Morris, Vol. 2 Num. 2, 2006

Related work:

- G. D. Plotkin: Call-by-name, call-by-value and the lambda calculus. Theoret. Comput. Sci., 1975.
- E. Machkasova, F. Turbak: A calculus for link-time compilation. ESOP 2000
- J. B. Wells, Detlef Plump, and Fairouz Kamareddine: Diagrams for meaning preservation. RTA 2003
- M. Schmidt-Schauß: Correctness of copy in calculi with letrec. RTA 2007.