Program Generation by Linking

Eugenio Moggi

moggi@disi.unige.it

DISI, Univ. of Genova

Goal

Explain multi-stage programming constructs (a la MetaML) in terms of more primitive concepts:

- names (labels) as used in programming in-the-large (record calculi)
- fresh name generation (gensym) a la FreshML

in the context of a monadic metalanguage

Related issues

- open code / $\lambda^{\bigcirc}$ / MetaML versus closed code / $\lambda^{\Box}$ / $\nu^{\Box}$
- symbolic evaluation of open code versus normal evaluation of programs

Formal results

- a 2-level version of MetaML: MetaML_2
- a CBV monadic translation $[\![-]\!]\colon \mathsf{MetaML}_2 \longrightarrow \mathsf{MML}_{\nu}^{N}$
- preservation of evaluation by $[\![-]\!]$ (no claim about typing)

A 2-level version of MetaML

A 2-level version of MetaML: MetaML_2

Syntax and semantics stratified in two levels: meta-level 0, object-level 1.

- $$\begin{array}{lll} p^0 \in \mathsf{P}^0 ::= & \underline{v^0} \mid p_1^0\ p_2^0 \mid \mathit{run}\ p^0 \mid \langle p^1 \rangle \\ v^0 \in \mathsf{V}^0 ::= & x^0 \mid \lambda x^0.p^0 \mid \langle v^1 \rangle \\ p^1 \in \mathsf{P}^1 ::= & \underline{v^1} \mid \lambda x^1.p^1 \mid p_1^1\ p_2^1 \mid \tilde{}p^0 \\ v^1 \in \mathsf{V}^1 ::= & x^1 \mid \lambda x^1.v^1 \mid v_1^1\ v_2^1 \mid \%\ v^0 \end{array}$$

  programs p and values v

  CBV λ-calculus fragment

- $$\begin{array}{lll} t^0 \in \mathsf{T}^0 ::= & b \mid t_1^0 \rightarrow t_2^0 \mid \langle t^1 \rangle \\ t^1 \in \mathsf{T}^1 ::= & b \mid t_1^1 \rightarrow t_2^1 \end{array}$$

  types t

  $\mathsf{T}^1 \subset \mathsf{T}^0$

A 2-level version of MetaML: design choices and informal semantics

Syntax and semantics stratified in two levels: meta-level 0, object-level 1.

- $$\begin{array}{lll} p^0 \in \mathsf{P}^0 ::= & \underline{v^0} \mid p_1^0\ p_2^0 \mid \mathit{run}\ p^0 \mid \langle p^1 \rangle \\ v^0 \in \mathsf{V}^0 ::= & x^0 \mid \lambda x^0.p^0 \mid \langle v^1 \rangle \\ p^1 \in \mathsf{P}^1 ::= & \underline{v^1} \mid \lambda x^1.p^1 \mid p_1^1\ p_2^1 \mid \tilde{}p^0 \\ v^1 \in \mathsf{V}^1 ::= & x^1 \mid \lambda x^1.v^1 \mid v_1^1\ v_2^1 \mid \%\ v^0 \end{array}$$

  programs p and values v

  CBV λ-calculus fragment

- values are explicitly marked, thus they don't have to be re-evaluated
- v^1 is an object program, while p^1 requires some meta-computation to get a v^1
- $\langle p^1 \rangle$ and $\tilde{}p^0$ allow to move between code $\langle v^1 \rangle$ and the corresponding v^1
- *run* p^0 evaluates p^0 to code $\langle v^1 \rangle$, then evaluates compiled object program v^1
- cross-stage persistence $\%\ v^0$ excluded to get simpler compilation (demotion)

A 2-level version of MetaML: evaluation relations $p^n \subset\!\!\xrightarrow{n} v^n$

Syntax and semantics stratified in two levels: meta-level 0, object-level 1.

$$
\begin{array}{lll}
p^0 \in \mathsf{P}^0 ::= & \underline{v^0} \mid p_1^0\ p_2^0 \mid \mathit{run}\ p^0 \mid \langle p^1 \rangle & \text{programs } p \text{ and} \\
v^0 \in \mathsf{V}^0 ::= & x^0 \mid \lambda x^0.p^0 \mid \langle v^1 \rangle & \text{values } v \\
p^1 \in \mathsf{P}^1 ::= & \underline{v^1} \mid \lambda x^1.p^1 \mid p_1^1\ p_2^1 \mid \tilde{}p^0 & \text{CBV } \lambda\text{-calculus fragment} \\
v^1 \in \mathsf{V}^1 ::= & x^1 \mid \lambda x^1.v^1 \mid v_1^1\ v_2^1 \mid \,\%\, v^0 &
\end{array}
$$

Compilation $x^1 \downarrow = \underline{x^0} \qquad (\lambda x^1.v^1)\downarrow = \underline{\lambda x^0.v^1 \downarrow} \qquad (v_1^1\ v_2^1)\downarrow = v_1^1 \downarrow\ v_2^1 \downarrow$

$$\frac{}{\underline{v^n} \subset\!\!\xrightarrow{n} v^n} \qquad \frac{p_1^0 \subset\!\!\xrightarrow{0} \lambda x^0.p^0 \quad p_2^0 \subset\!\!\xrightarrow{0} v^0 \quad p^0[x^0 : v^0] \subset\!\!\xrightarrow{0} v_1^0}{p_1^0\ p_2^0 \subset\!\!\xrightarrow{0} v_1^0}$$

$$\frac{p^0 \subset\!\!\xrightarrow{0} \langle v^1 \rangle \quad \boxed{v^1 \downarrow} \subset\!\!\xrightarrow{0} v^0}{\mathit{run}\ p^0 \subset\!\!\xrightarrow{0} v^0}\ \mathrm{FV}(v^1) = \emptyset \qquad \frac{p^1 \subset\!\!\xrightarrow{1} v^1}{\langle p^1 \rangle \subset\!\!\xrightarrow{0} \langle v^1 \rangle}$$

$$\frac{p^1 \subset\!\!\xrightarrow{1} v^1}{\lambda x^1.p^1 \subset\!\!\xrightarrow{1} \lambda x^1.v^1} \qquad \frac{p_1^1 \subset\!\!\xrightarrow{1} v_1^1 \quad p_2^1 \subset\!\!\xrightarrow{1} v_2^1}{p_1^1\ p_2^1 \subset\!\!\xrightarrow{1} v_1^1\ v_2^1} \qquad \frac{p^0 \subset\!\!\xrightarrow{0} \langle v^1 \rangle}{\tilde{}p^0 \subset\!\!\xrightarrow{1} v^1}$$

A Monadic Metalanguage with Names

Monadic Metalanguage: general ideas

- $e \in \mathsf{E} ::= x \mid \lambda x.e \mid e_1\ e_2 \mid \ldots \mid$ *ret* $e \mid$ *do* $e_1\ e_2 \mid \ldots$ terms
- $\tau \in \mathsf{T} ::= b \mid \tau_1 \to \tau_2 \mid M\tau \mid \ldots$ types
- Monadic metalanguages mediate between programming languages and λ-calculi/type theories for maths and proofs

  computational types $M\tau$ make explicit distinction values/computations
- ChAM-like (Chemical Abstract Machine) operational semantics:
  - *transparent* simplification $\longrightarrow$ – as in (pure) calculi
  - *programmable* computation $\longmapsto$ – as in programming languages

  computation not bundled with a deterministic simplification strategy

  Addition of other computational effects *easy*!
- Programming languages semantics via translation to monadic metalanguage

Monadic Metalanguage: MML^N_ν

- $e \in \mathsf{E} ::= x \mid \lambda x.e \mid e_1\ e_2 \mid \theta.X \mid \mathit{b}(r)e \mid e\langle\theta\rangle \mid$ *ret* $e \mid$ *do* $e_1\ e_2 \mid \nu X.e$ terms
- $\theta \in \mathsf{ER} ::= r \mid ? \mid \theta\{X : e\}$ name resolvers

$X \in \mathsf{N}$ symbolic name, $x \in \mathsf{X}$ term variable, $r \in \mathsf{R}$ resolver variable

Commentary to BNF

name resolver) θ denotes partial function $\mathsf{N} \xrightarrow{fin} \mathsf{E}$ from names to terms

name resolution) $\theta.X$ term obtained when θ resolves X

fragment) $\mathit{b}(r)e$ denotes function $(\mathsf{N} \xrightarrow{fin} \mathsf{E}) \to \mathsf{E}$ from resolvers to terms

linking) $e\langle\theta\rangle$ term obtained when fragment e is linked to r

Monadic Metalanguage: MML_{ν}^{N}

- $e \in \mathsf{E}::= x \mid \lambda x.e \mid e_1\ e_2 \mid \theta.X \mid \mathsf{b}(r)e \mid e\langle\theta\rangle \mid$ *ret* $e \mid$ *do* $e_1\ e_2 \mid \nu X.e$ terms
- $\theta \in \mathsf{ER}::= r \mid ? \mid \theta\{X:e\}$ name resolvers

Operational Semantics for Monadic Metalanguages

- simplification $e \longrightarrow e'$ confluent relation defined as compatible closure
- computation $Id \longmapsto Id'$ describing how *configurations* may evolve
- name resolvers θ as extensible records
  - resolvers are handled by simplification
  - calculus is *expressive* even with *second-class* resolvers
- name generation $\nu X.e$ is a computational effect, as in FreshML [SGP03]
  - name generation *essential* to prevent accidental overriding of resolver
  - object language syntax modulo α-conversion (as in FreshML) not our aim!
  - MML_{ν}^{N} supports program generation not program analysis/transformation

Operational Semantics of MML^N_ν: Simplification

- $e \in \mathsf{E}::= \quad x \mid \lambda x.e \mid e_1\ e_2 \mid \theta.X \mid \mathit{b}(r)e \mid e\langle\theta\rangle \mid \mathit{ret}\ e \mid \mathit{do}\ e_1\ e_2 \mid \nu X.e$
- $\theta \in \mathsf{ER}::= r \mid ? \mid \theta\{X:e\}$

beta) $(\lambda x.e_2)\ e_1 \longrightarrow e_2[x:e_1]$

resolve) $(\theta\{X:e\}).X \longrightarrow e$

delegate) $(\theta\{X:e\}).X' \longrightarrow \theta.X'$ if $X' \neq X$

link) $(\mathit{b}(r)e)\langle\theta\rangle \longrightarrow e[r:\theta]$

Prop. The simplification relation $\longrightarrow$ is confluent.

Operational Semantics of MML^N_ν: Computation

- $e \in \mathsf{E} ::= x \mid \lambda x.e \mid e_1\ e_2 \mid \theta.X \mid \mathsf{b}(r)e \mid e\langle\theta\rangle \mid \mathit{ret}\ e \mid \mathit{do}\ e_1\ e_2 \mid \nu X.e$
- $\theta \in \mathsf{ER} ::= r \mid ? \mid \theta\{X:e\}$
- $Id \equiv (\mathcal{X}|e, E)$ configuration: $\mathcal{X} \subseteq_{fin}$ N current name space,
  $e \in$ E program fragment under consideration, $E \in \mathsf{E}^*$ stack of things to do
- Administrative steps

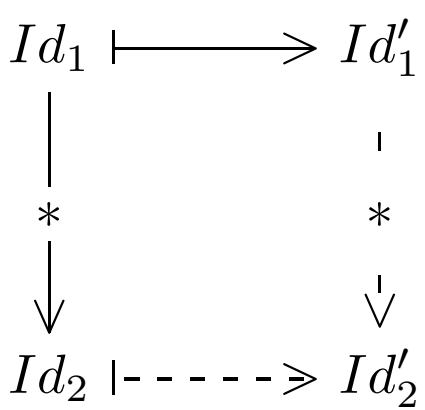
(push) $(\mathcal{X}|\mathit{do}\ e_1\ e_2, E) \longmapsto (\mathcal{X}|e_1, e_2 :: E)$

(pop) $(\mathcal{X}|\mathit{ret}\ e_1, e_2 :: E) \longmapsto (\mathcal{X}|e_2\ e_1, E)$

- Name generation step

(ν) $(\mathcal{X}|\nu X.e, E) \longmapsto (\mathcal{X}, X|e, E)$ with X **renamed to avoid clashes**, i.e. $X \notin \mathcal{X}$

Prop. Computation is insensitive to further simplification



Monadic Translation and its Properties

CBV monadic translation $[\![-]\!]\colon \mathsf{MetaML}_1 \longrightarrow \mathsf{MML}$

- $[\![-]\!]^\rho$ with ρ defined on $\mathrm{FV}(-)$ and $e_x^0 \stackrel{\Delta}{\equiv} \rho(x^0) \in \mathsf{E}$
  (the format $[\![-]\!]^\rho$ is similar to an interpretation function)

$v^0 \colon t^0$	$[\![v^0]\!]^\rho \colon \tau \in \mathsf{E}$
x^0	$e_x^0 = \rho(x^0)$
$\lambda x^0.p^0$	$\lambda x.[\![p^0]\!]^{\rho, x^0 : x}$

$p^0 \colon t^0$	$e^0 \equiv [\![p^0]\!]^\rho \colon M\tau \in \mathsf{E}$
$\underline{v^0}$	*ret* $[\![v^0]\!]^\rho$
$p_1^0 p_2^0$	*do* $x_1 \leftarrow e_1^0; x_2 \leftarrow e_2^0;\ x_1 x_2$

  we use Haskell's do-notation *do* $x \leftarrow e_1;\ e_2 \stackrel{\Delta}{\equiv}$ *do* $e_1\ (\lambda x.e_2)$

- Substitution lemma – the translation commutes with substitution $-[x^0 \colon v^0]$
  $[\![\_[x^0 \colon v^0]]\!]^\rho = [\![\_]\!]^{\rho, x^0 : e}$, if $e = [\![v^0]\!]^\rho$

- The translation preserves the evaluation relation $p^0 \stackrel{0}{\subset\!\longrightarrow} v^0$
  If $p^0 \stackrel{0}{\subset\!\longrightarrow} v^0$, then $([\![p^0]\!]^\rho, E) \stackrel{*}{\Longrightarrow} (\textit{ret}\ [\![v^0]\!]^\rho, E)$ for any ρ and E
  $\Longrightarrow \stackrel{\Delta}{\equiv} \longrightarrow \cup \longmapsto$ is simplification+computation on configurations

CBV monadic translation $[\![-]\!]\colon \mathsf{MetaML}_2 \longrightarrow \mathsf{MML}_\nu^N$

- $[\![-]\!]^\rho$ with ρ defined on $\mathrm{FV}(-)$ and $e_x^0 \stackrel{\Delta}{\equiv} \rho(x^0)$, $\boxed{b(r)e_x^1 \stackrel{\Delta}{\equiv} \rho(x^1) \in \mathsf{E}}$

  $[\![v^0]\!]^\rho\colon \tau \in \mathsf{E} \quad [\![p^0]\!]^\rho\colon M\tau \in \mathsf{E}$

  $\boxed{[\![p^1]\!]^\rho\colon M[\ldots|M\tau] \in \mathsf{E} \quad [\![v^1]\!]_\theta^\rho\colon M\tau \in \mathsf{E} \text{ where } \theta \in \mathsf{ER} \text{ name resolver}}$

- Substitution lemma – the translation commutes with substitution $-[x^0\colon v^0]$

  $[\![\_[x^0\colon v^0]]\!]^\rho = [\![\_]\!]^{\rho,x^0:e}$ and $[\![v^1[x^0\colon v^0]]\!]_\theta^\rho = [\![v^1]\!]_\theta^{\rho,x^0:e}$, if $e = [\![v^0]\!]^\rho$

- Simplification lemma – technical property specific to the translation of v^1

  $[\![v^1]\!]_{\theta_1}^{\rho_1} \stackrel{*}{\longrightarrow} [\![v^1]\!]_{\theta_2}^{\rho_2}$, if $e_1[r\colon\theta_1] \stackrel{*}{\longrightarrow} e_2[r\colon\theta_2]$ when $\rho_i(x^1) = b(r)e_i$

- Compilation lemma – relate translations of v^1 and $p^0 \equiv v^1\downarrow$

  $[\![v^1]\!]_\theta^\rho = [\![p^0]\!]^{\rho'}$, if $\rho'(x^0) = e[r\colon\theta]$ whenever $\rho(x^1) = b(r)e$

- The translation preserves the evaluation relations $p^n \stackrel{n}{\subset\!\longrightarrow} v^n$
  - If $p^0 \stackrel{0}{\subset\!\longrightarrow} v^0$, then $(\mathcal{X}|[\![p^0]\!]^\rho, E) \stackrel{*}{\Longrightarrow} (\mathcal{X}'|\mathit{ret}\ [\![v^0]\!]^\rho, E)$ for some $\mathcal{X}' \supseteq \mathcal{X}$
  - If $p^1 \stackrel{1}{\subset\!\longrightarrow} v^1$, then $(\mathcal{X}|[\![p^1]\!]^\rho, E) \stackrel{*}{\Longrightarrow} (\mathcal{X}'|\mathit{ret}\ b(r)[\![v^1]\!]_r^\rho, E)$ for some $\mathcal{X}' \supseteq \mathcal{X}$

  $\Longrightarrow \stackrel{\Delta}{\equiv} \longrightarrow \cup \longmapsto$ is simplification+computation on configurations

CBV monadic translation $[\![-]\!]\colon \mathsf{MetaML}_2 \longrightarrow \mathsf{MML}_\nu^N$

- $[\![-]\!]^\rho$ with ρ defined on $\mathrm{FV}(-)$ and $e_x^0 \triangleq \rho(x^0)$, $b(r)e_x^1 \triangleq \rho(x^1) \in \mathsf{E}$

v^0	$[\![v^0]\!]^\rho \in \mathsf{E}$	p^0	$e^0 \equiv [\![p^0]\!]^\rho \in \mathsf{E}$
	...		...
$\langle v^1 \rangle$	$b(r)[\![v^1]\!]_r^\rho$	*run* p^0	*do* $x \leftarrow e^0;\ x\langle ? \rangle$

extends MetaML_1 translation

$v^1 : t^1$	$[\![v^1]\!]_\theta^\rho : M\tau \in \mathsf{E}$ where $\theta \in \mathsf{ER}$ name resolver
x^1	*ret* $e_x^1[r:\theta]$ where $b(r)e_x^1 = \rho(x^1)$
$\lambda x^1.v^1$	*ret* $\lambda x.[\![v^1]\!]_\theta^{\rho, x^1 : b(r)x}$
$v_1^1 v_2^1$	*do* $x_1 \leftarrow [\![v_1^1]\!]_\theta^\rho; x_2 \leftarrow [\![v_2^1]\!]_\theta^\rho;\ x_1 x_2$
$p^1 : t^1$	$e^1 \equiv [\![p^1]\!]^\rho : M[\ldots \mid M\tau] \in \mathsf{E}$
$\underline{v^1}$	*ret* $(b(r)[\![v^1]\!]_r^\rho)$
$p_1^1 p_2^1$	*do* $x_1' \leftarrow e_1^1; x_2' \leftarrow e_2^1;$ *ret* $(b(r)$*do* $x_1 \leftarrow x_1'\langle r \rangle; x_2 \leftarrow x_2'\langle r \rangle;\ x_1 x_2)$
$\lambda x^1.p^1$	$\nu X.$*do* $x' \leftarrow [\![p^1]\!]^{\rho, x^1 : b(r)r.X};$ *ret* $(b(r)$*ret* $\lambda x.x'\langle r\{X : x\}\rangle)$

trivial cases: $[\![\tilde{} p^0]\!]^\rho = [\![p^0]\!]^\rho$ and $[\![\langle p^1 \rangle]\!]^\rho = [\![p^1]\!]^\rho$

- cross-stage persistence: $[\![\%\ v^0]\!]_\theta^\rho =$ *ret* $[\![v^0]\!]^\rho$ is insensitive to θ

CBV monadic translation $[\![-]\!]\colon \mathsf{MetaML}_2 \longrightarrow \mathsf{MML}_\nu^N$

Examples of evaluation and translation

- Value $v^0 \stackrel{\Delta}{\equiv} \langle \lambda x^1.x^1 \rangle$ is code representing the object-level identity function
  $e_0 \stackrel{\Delta}{=} [\![v^0]\!]$ is $\mathit{b}(r)\mathit{ret}\ \lambda x.\mathit{ret}\ x$ – body is CBV translation of identity function
- Program $p_1^0 \stackrel{\Delta}{\equiv} \underline{v^0}$ evaluates immediately to v^0. The translation $[\![p_1^0]\!]^\emptyset$ is $\mathit{ret}\ e_0$, thus preservation of $p_1^0 \subset\!\!\xrightarrow{0} v^0$ holds trivially
- Program $p_2^0 \stackrel{\Delta}{\equiv} \langle \lambda x^1.\underline{x^1} \rangle$ evaluates to v^0 too, but the evaluation is more complex. as mirrored by the translation $[\![p_2^0]\!]^\emptyset$

$$\nu X.\ \mathit{do}\ x' \leftarrow \mathit{ret}\ (\mathit{b}(r)\mathit{ret}\ r.X);\ \ \mathit{ret}\ (\mathit{b}(r)\mathit{ret}\ \lambda x.x'\langle r\{X\colon x\}\rangle)$$

and preservation of $p_2^0 \subset\!\!\xrightarrow{0} v^0$ takes the following form

1. $(\emptyset|e_2^0,[]) \stackrel{+}{\Longrightarrow}$ one name generation and some administrative steps
2. $(X|\mathit{ret}\ (\mathit{b}(r)\mathit{ret}\ \lambda x.\boxed{(\mathit{b}(r)\mathit{ret}\ r.X)\langle r\{X\colon x\}\rangle}),[]) \stackrel{+}{\longrightarrow}$ simplification lemma
3. $(X|\mathit{ret}\ \mathit{b}(r)\mathit{ret}\ \lambda x.\mathit{ret}\ x,[]) \equiv (X|\mathit{ret}\ e_0,[])$