Full reduction at full throttle
Section 1

Maxime Klusman

Radboud University Nijmegen

January 23, 2015

Recap

$$\mathbf{Terms} \ni t ::= x \mid t_1\ t_2 \mid v$$
$$\mathbf{Val} \ni v ::= \lambda x.t \mid [\tilde{x}\ v_1 ... v_n]$$

$$(\lambda x.t)\ v \to t[x \leftarrow v] \qquad (\beta_v)$$
$$[\tilde{x}\ v_1 ... v_n]\ v \to [\tilde{x}\ v_1 ... v_n\ v] \qquad (\beta_s)$$
$$\Gamma(t) \to \Gamma(t') \qquad \text{if } t \to t' \qquad \text{context}$$

$$\Gamma ::= t[\,]|[\,]v$$

$$\mathcal{N}(t) = \mathcal{R}(\mathcal{V}(t)) \qquad (1)$$
$$\mathcal{R}(\lambda x.t) = \lambda y.\mathcal{N}((\lambda x.t)\ [\tilde{y}]) \quad \langle y \text{ fresh}\rangle \qquad (2)$$
$$\mathcal{R}[\tilde{x}\ v_1 ... v_n] = x\ \mathcal{R}(v_1) ... \mathcal{R}(v_n) \qquad (3)$$

Abstract setting

```
module type Values = sig
  type t
  val app : t -> t -> t
  type atom =
    | Var of var
  type head =
    | Lam of t -> t
    | Accu of atom * t list
  val head : t -> head
  val mkLam : (t -> t) -> t
  val mkAccu : atom -> t
end
```

Compilation and normalization

$$\ulcorner x \urcorner^B = \begin{cases} x & \text{if } x \in B \\ \text{mkAccu}(\text{Var } x) & \text{otherwise} \end{cases}$$

$$\ulcorner \lambda x.t \urcorner^B = \text{mkLam}(\text{fun } x \to \ulcorner t \urcorner^{B \cup \{x\}})$$

$$\ulcorner t_1 t_2 \urcorner^B = \text{app} \ulcorner t_1 \urcorner^B \ulcorner t_2 \urcorner^B$$

$$\mathcal{N}_\Lambda \; t = \mathcal{R}_V \; \ulcorner t \urcorner^\emptyset$$

$$\mathcal{R}_V \; v = \mathcal{R} \; (\text{head } v)$$

$$\mathcal{R} \; (\text{Lam } f) = \lambda y. \mathcal{R}_V \; (f(\text{mkAccu}(\text{Var } y))) \quad \langle y \text{ fresh} \rangle$$

$$\mathcal{R} \; (\text{Accu}(a, [v_n; ...; v_1])) = (\mathcal{R}_A \; a)(\mathcal{R}_V \; v_1)...(\mathcal{R}_V \; v_n)$$

$$\mathcal{R}_A \; (\text{Var } x) = x$$

Tagged normalization

```
type t = head
let app t v = match t with
  | Lam f -> f v
  | Accu(a, args) -> (Accu(a, v::args))
let head v = v
let mkLam f = Lam f
let mkAccu a = Accu(a,[])
```

Example (tagged)

$$\begin{aligned}
\mathcal{N}_\Lambda\ t &=_1 \mathcal{R}_V\ \ulcorner t \urcorner^\emptyset \\
\mathcal{R}_V\ v &=_2 \mathcal{R}\ (\text{head}\ v) \\
\mathcal{R}\ (\text{Lam}\ f) &=_3 \lambda y.\mathcal{R}_V\ (f(\text{mkAccu}(\text{Var}\ y))) \quad \langle y \text{ fresh}\rangle \\
\mathcal{R}\ (\text{Accu}(a, [v_n; ...; v_1])) &=_4 (\mathcal{R}_A\ a)(\mathcal{R}_V\ v_1)...(\mathcal{R}_V\ v_n) \\
\mathcal{R}_A\ (\text{Var}\ x) &=_5 x
\end{aligned}$$

$$\begin{aligned}
\mathcal{N}_\Lambda\ t &=_1 \mathcal{R}_V\ \ulcorner t \urcorner^\emptyset = \mathcal{R}_V(\text{Lam}\ f_1) \\
&=_2 \mathcal{R}\ (\text{head}\ (\text{Lam}\ f_1)) = \mathcal{R}\ (\text{Lam}\ f_1) \\
&=_3 \lambda y.\mathcal{R}_V\ (f_1(\text{mkAccu}\ (\text{Var}\ y))) = \lambda y.\mathcal{R}_V\ (\text{Lam}\ f_2) \\
(=_2=)\ &=_3 \lambda y.\lambda z.\mathcal{R}_V\ (f_2(\text{mkAccu}\ (\text{Var}\ z))) = \lambda y.\lambda z.\mathcal{R}_V\ (\text{Accu}\ (\text{Var y}, v :: [\,])) \\
(=_2=)\ &=_4 \lambda y.\lambda z.(\mathcal{R}_A(\text{Var}\ y))\ (\mathcal{R}_V\ v) = \lambda y.\lambda z.(\mathcal{R}_A(\text{Var}\ y))\ (\mathcal{R}_V\ \text{Accu}(\text{Var}\ z, [\,])) \\
&=_5 \lambda y.\lambda z.y\ (\mathcal{R}_V\ \text{Accu}(\text{Var}\ z, [\,])) \\
(=_2=)\ &=_4 \lambda y.\lambda z.y\ (\mathcal{R}_A(\text{Var}\ z)) \\
&=_5 \lambda y.\lambda z.y\ z
\end{aligned}$$

Tagless normalization: memory representation

Closures:

```
+----------+-------+----------+------+-------+     +-------+
|   size   | color |    tag   |   C  |   v1  | ... |   v2  |
+----------+-------+----------+------+-------+     +-------+
\__________________________/ \_____/ \__________________/
          header              code ptr   values of free vars
```

```
let g = let x = 4 and y = 3 in
        fun x y z -> x + y + z
```

```
+----------+-------+----------+------+-------+-------+
| size = 3 |  ...  | tag =/= 0 |  ... |   4   |   3   |
+----------+-------+----------+------+-------+-------+
```

Tagless normalization

```
type t = t -> t
let app f v = f v

let getAtom o = (Obj.magic (Obj.field o 3)) : atom
let getArgs o = (Obj.magic (Obj.field o 4)) : t list
let rec head (v:t) =
  let o = Obj.repr v in
  if Obj.tag o = 0 then Accu(getAtoms o, getArgs o) else Lam v

let mkLam f = f

let rec accu atom args =
  let res = fun v -> accu atom (v::args) in
  Obj.set_tag (Obj.repr res) 0; (res : t)
let mkAccu atom = accu atom []
```

Example (tagless)

$$\begin{aligned}
\mathcal{N}_\Lambda\ t &=_1 \mathcal{R}_V\ \ulcorner t \urcorner^{\emptyset} \\
\mathcal{R}_V\ v &=_2 \mathcal{R}\ (\text{head}\ v) \\
\mathcal{R}\ (\text{Lam}\ f) &=_3 \lambda y.\mathcal{R}_V\ (f(\text{mkAccu}(\text{Var}\ y))) \quad \langle y \text{ fresh}\rangle \\
\mathcal{R}\ (\text{Accu}(a, [v_n; ...; v_1])) &=_4 (\mathcal{R}_A\ a)(\mathcal{R}_V\ v_1)...(\mathcal{R}_V\ v_n) \\
\mathcal{R}_A\ (\text{Var}\ x) &=_5 x
\end{aligned}$$

$$\begin{aligned}
\mathcal{N}_\Lambda\ t &=_1 \mathcal{R}_V\ \ulcorner t \urcorner^{\emptyset} = \mathcal{R}_V\ f_1 =_2 \mathcal{R}\ (\text{head}\ f_1) = \mathcal{R}\ (\text{Lam}\ f_1) \\
&=_3 \lambda y.\mathcal{R}_V\ (f_1(\text{mkAccu}\ (\text{Var}\ y))) = \lambda y.\mathcal{R}_V\ f_2 \\
(=_2=)\ &=_3 \lambda y.\lambda z.\mathcal{R}_V\ (f_2(\text{mkAccu}\ (\text{Var}\ z))) = \lambda y.\lambda z.\mathcal{R}_V\ (\text{fun}\ u \underset{\text{tag}=0}{\rightarrow} b) \\
&=_2 \lambda y.\lambda z.\mathcal{R}\ (\text{head}\ (\text{fun}\ u \underset{\text{tag}=0}{\rightarrow} b)) = \lambda y.\lambda z.\mathcal{R}\ (\text{Accu}(\text{Var}\ y, w :: [\,])) \\
&=_4 \lambda y.\lambda z.(\mathcal{R}_A(\text{Var}\ y))\ (\mathcal{R}_V\ w) = \lambda y.\lambda z.(\mathcal{R}_A(\text{Var}\ y))\ (\mathcal{R}_V\ (\text{fun}\ v \underset{\text{tag}=0}{\rightarrow} c)) \\
&=_5 \lambda y.\lambda z.y\ (\mathcal{R}_V\ (\text{fun}\ v \underset{\text{tag}=0}{\rightarrow} c)) \\
&=_2 \lambda y.\lambda z.y\ (\mathcal{R}\ (\text{head}\ (\text{fun}\ v \underset{\text{tag}=0}{\rightarrow} c))) = \mathcal{R}\ (\text{Accu}(\text{Var}\ z, [\,])) \\
&=_4 \lambda y.\lambda z.y\ (\mathcal{R}_A(\text{Var}\ z)) =_5 \lambda y.\lambda z.y\ z
\end{aligned}$$

Questions?