Full reduction at full throttle
Section 1

Maxime Klusman

Radboud University Nijmegen

January 23, 2015
**Terms** \( \ni t ::= x \mid t_1 t_2 \mid v \)

**Val** \( \ni v ::= \lambda x.t \mid [\tilde{x} v_1...v_n] \)

\[(\lambda x.t) v \rightarrow t[x \leftarrow v] \quad (\beta_v)\]

\[[\tilde{x} v_1...v_n] v \rightarrow [\tilde{x} v_1...v_n v] \quad (\beta_s)\]

\[\Gamma(t) \rightarrow \Gamma(t') \quad \text{if } t \rightarrow t' \quad \text{context}\]

\[\Gamma ::= t[\emptyset][\emptyset]v\]

\[\mathcal{N}(t) = \mathcal{R}(\mathcal{V}(t)) \quad (1)\]

\[\mathcal{R}(\lambda x.t) = \lambda y.\mathcal{N}((\lambda x.t)[\tilde{y}]) \quad \langle y \text{ fresh} \rangle \quad (2)\]

\[\mathcal{R}[\tilde{x} v_1...v_n] = x \mathcal{R}(v_1)...\mathcal{R}(v_n) \quad (3)\]
Abstract setting

```ocaml
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

\[ \Gamma x \vdash B = \begin{cases} x & \text{if } x \in B \\ \text{mkAccu(Var } x) & \text{otherwise} \end{cases} \]

\[ \Gamma \lambda x.t \vdash B = \text{mkLam(fam } x \rightarrow \Gamma t \vdash B \cup \{x\}) \]

\[ \Gamma t_1 t_2 \vdash B = \text{app} \Gamma t_1 \vdash B \Gamma t_2 \vdash B \]

\[ \mathcal{N}_\Lambda t = \mathcal{R}_V \Gamma t \vdash \emptyset \]

\[ \mathcal{R}_V v = \mathcal{R}(\text{head } v) \]

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

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

\[ \mathcal{R}_A(\text{Var } x) = x \]
Tagged normalization

```ocaml
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)

\[\begin{align*}
N_\Lambda \ t &= 1 \mathcal{R}_V \ t^\emptyset \\
\mathcal{R}_V \ v &= 2 \mathcal{R} \ (\text{head } v) \\
\mathcal{R} \ (\text{Lam } f) &= 3 \lambda y.\mathcal{R}_V \ (f(\text{mkAccu}(\text{Var } y))) \quad \langle y \text{ fresh} \rangle \\
\mathcal{R} \ (\text{Accu}(a, [v_n; \ldots; v_1])) &= 4 (\mathcal{R}_A a)(\mathcal{R}_V v_1)\ldots(\mathcal{R}_V v_n) \\
\mathcal{R}_A \ (\text{Var } x) &= 5 x
\end{align*}\]

\[\begin{align*}
N_\Lambda \ t &= 1 \mathcal{R}_V \ t^\emptyset = \mathcal{R}_V (\text{Lam } f_1) \\
&= 2 \mathcal{R} \ (\text{head } (\text{Lam } f_1)) = \mathcal{R} \ (\text{Lam } f_1) \\
&= 3 \lambda y.\mathcal{R}_V \ (f_1(\text{mkAccu}(\text{Var } y))) = \lambda y.\mathcal{R}_V \ (\text{Lam } f_2) \\
(=_{2=}) &= 3 \lambda y.\lambda z.\mathcal{R}_V \ (f_2(\text{mkAccu}(\text{Var } z))) = \lambda y.\lambda z.\mathcal{R}_V \ (\text{Accu}(\text{Var } y, v :: [])) \\
(=_{2=}) &= 4 \lambda y.\lambda z.(\mathcal{R}_A (\text{Var } y))(\mathcal{R}_V v) = \lambda y.\lambda z.(\mathcal{R}_A (\text{Var } y))(\mathcal{R}_V \text{Accu}(\text{Var } z, [])) \\
&= 5 \lambda y.\lambda z.y \ (\mathcal{R}_V \text{Accu}(\text{Var } z, [])) \\
(=_{2=}) &= 4 \lambda y.\lambda z.y \ (\mathcal{R}_A (\text{Var } z)) \\
&= 5 \lambda y.\lambda z.y \ z
\end{align*}\]
Tagless normalization: memory representation

Closures:

<table>
<thead>
<tr>
<th>size</th>
<th>color</th>
<th>tag</th>
<th>C</th>
<th>v1</th>
<th>...</th>
<th>v2</th>
</tr>
</thead>
</table>

\--------------------------/ \_____/ \___________________/
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

```ocaml
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)

\[ N_{\Lambda} t =_1 R_V \downarrow t^{\emptyset} \]

\[ R_V v =_2 R \text{ (head } v) \]

\[ R \text{ (Lam } f) =_3 \lambda y. R_V (f(\text{mkAccu}(\text{Var } y))) \quad \langle y \text{ fresh} \rangle \]

\[ R \text{ (Accu}(a, [v_n; \ldots; v_1])) =_4 (R_A a)(R_V v_1) \ldots (R_V v_n) \]

\[ R_A (\text{Var } x) =_5 x \]

\[ N_{\Lambda} t =_1 R_V \downarrow t^{\emptyset} = R_V f_1 =_2 R \text{ (head } f_1) = R \text{ (Lam } f_1) \]

\[ =_3 \lambda y. R_V (f_1(\text{mkAccu}(\text{Var } y))) = \lambda y. R_V f_2 \]

\[ (=_2=) =_3 \lambda y. \lambda z. R_V (f_2(\text{mkAccu}(\text{Var } z))) = \lambda y. \lambda z. R_V \text{ (fun } u \rightarrow \text{ tag=0 } b) \]

\[ =_2 \lambda y. \lambda z. R \text{ (head (fun } u \rightarrow \text{ tag=0 } b)) = \lambda y. \lambda z. R \text{ (Accu}(\text{Var } y, w :: [])) \]

\[ =_4 \lambda y. \lambda z. (R_A(\text{Var } y))(R_V w) = \lambda y. \lambda z. (R_A(\text{Var } y))(R_V \text{ (fun } v \rightarrow \text{ tag=0 } c)) \]

\[ =_5 \lambda y. \lambda z. y (R_V \text{ (fun } v \rightarrow \text{ tag=0 } c)) \]

\[ =_2 \lambda y. \lambda z. y (R \text{ (head (fun } v \rightarrow \text{ tag=0 } c))) = R \text{ (Accu}(\text{Var } z, [])) \]

\[ =_4 \lambda y. \lambda z. y (R_A(\text{Var } z)) =_5 \lambda y. \lambda z. y z \]
Questions?