Efficient normalization by evaluation

Mathieu Boespflug

École Polytechnique

15 August 2009

Convertibility

$$t_1 \rightarrow_\beta \ldots \leftarrow_\beta \ldots \rightarrow_\beta \ldots \leftarrow_\beta t_2$$

Convertibility

$$\begin{array}{ccc} t_1 & & t_2 \\ \downarrow^*_\beta & & \downarrow^*_\beta \\ t'_1 & \stackrel{?}{\equiv} & t'_2 \end{array}$$

$$M \quad : \qquad 55 < 1337$$

$$M : \quad 55 < 1337$$

$$M : (\lambda x.\, 55)\ 10 < 1337$$

$$
\begin{aligned}
M &: \quad 55 < 1337 \\
M &: \quad (\lambda x.\, 55)\; 10 < 1337 \\
M &: \quad \mathit{fib}\; 10 < 1337
\end{aligned}
$$

The conversion test

$$\frac{\Gamma \vdash t : A \quad A \equiv_\beta B}{\Gamma \vdash t : B}$$

Seek:

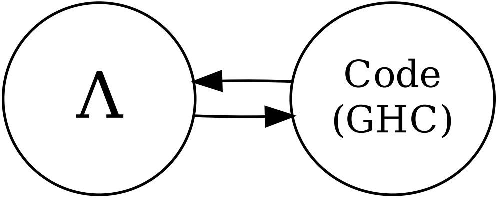
- simplicity
- efficiency

fast

fast cheap

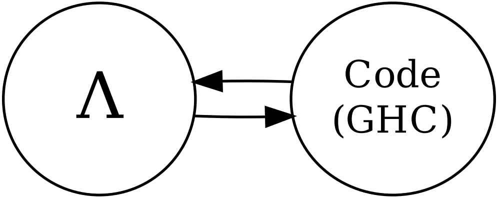
fast cheap general

General overview



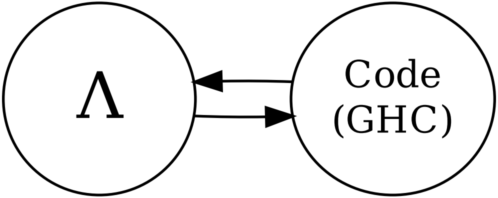
- Plenty of existing (fast) reduction devices.

General overview



- Plenty of existing (fast) reduction devices.
- Solution: reuse them!

General overview



- Plenty of existing (fast) reduction devices.
- Solution: reuse them!
- Advantage: separation of concerns.

Interpretation of terms

$$
\begin{aligned}
[\![\underline{x}]\!]\ n &= \hat{x} && \text{if } x < n \\
[\![\underline{x}]\!]\ n &= \mathit{Con}\ \underline{x} && \text{otherwise} \\
[\![\underline{\lambda x.\ t}]\!]\ n &= \mathit{Abs}\ (\lambda \hat{n} \to [\![t]\!]\ (n+1)) \\
[\![\underline{t_1\ t_2}]\!]\ n &= \mathit{App}\ ([\![t_1]\!]\ n)\ ([\![t_2]\!]\ n)
\end{aligned}
$$

Interpretation of terms (example)

$$[\![\underline{(\lambda x.\,(\lambda y.\; y\; x))\; z}]\!]$$

Interpretation of terms (example)

$$\frac{[\![\underline{(\lambda x.\ (\lambda y.\ y\ x))\ z}]\!]}{ap\ (Abs\ (\lambda x \to Abs\ (\lambda y \to ap\ y\ x)))\ (Con\ \texttt{"0"})}$$

A simple HOAS normalizer

$$
\begin{array}{l}
\mathit{norm}\ n\ (\mathit{App}\ t_1\ t_2) = \\
\quad \mathbf{case}\ \mathit{norm}\ n\ t_1\ \mathbf{of} \\
\quad\quad \mathit{Abs}\ t_1' \rightarrow \mathit{norm}\ n\ (t_1'\ t_2) \\
\quad\quad t_1' \rightarrow t_1'\ \underline{@}\ (\mathit{norm}\ n\ t_2) \\
\mathit{norm}\ n\ (\mathit{Abs}\ t) = \\
\quad \underline{\lambda.}\ (\mathit{norm}\ (n+1)\ (t\ (\mathit{Con}\ (\mathit{show}\ n)))) \\
\mathit{norm}\ n\ (\mathit{Con}\ c) = \underline{c}
\end{array}
$$

Towards normalization by evaluation

$$
\begin{array}{l}
ap\ t_1\ t_2 = \textbf{case}\ norm\ n\ t_1\ \textbf{of} \\
\quad Abs\ t_1' \to norm\ n\ (t_1'\ t_2) \\
\quad t_1' \to t_1'\ \underline{@}\ (norm\ n\ t_2)
\end{array}
$$

$$
\begin{array}{l}
norm\ n\ (App\ t_1\ t_2) = ap\ t_1\ t_2 \\
norm\ n\ (Abs\ t) = \\
\quad \underline{\lambda.}\ (norm\ (n+1)\ (t\ (Con\ (show\ n)))) \\
norm\ n\ (Con\ c) = \underline{c}
\end{array}
$$

Interpretation of terms (revised)

$$
\begin{aligned}
[\![\underline{x}]\!]\ n &= \hat{x} && \text{if } x < n \\
[\![\underline{x}]\!]\ n &= \mathit{Con}\ \underline{x} && \text{otherwise} \\
[\![\underline{\lambda x.\ t}]\!]\ n &= \mathit{Abs}\ (\lambda \hat{n} \to [\![t]\!]\ (n+1)) \\
[\![\underline{t_1\ t_2}]\!]\ n &= \mathit{ap}\ ([\![t_1]\!]\ n)\ ([\![t_2]\!]\ n)
\end{aligned}
$$

Normalizer (revised)

$$ap\ (Abs\ f)\ t = f\ t$$
$$ap\ t_1\ t_2 = t_1\ \underline{@}\ t_2$$

$$norm\ n\ (Abs\ t) =$$
$$\quad \underline{\lambda.}\ (norm\ (n+1)\ (t\ (Con\ (show\ n))))$$
$$norm\ n\ (Con\ c) = \underline{c}$$

Optimizations

Intermediate closures (example)

$$
\begin{aligned}
&[\![map\ id\ nil]\!] \\
&\qquad = ap\ (ap\ map\ id)\ nil
\end{aligned}
$$

Intermediate closures (example)

$$
\begin{aligned}
&[\![\mathit{map}\ \mathit{id}\ \mathit{nil}]\!] \\
&\qquad = \mathit{ap}\ (\mathit{ap}\ \mathit{map}\ \mathit{id})\ \mathit{nil} \\
&\qquad = \mathit{ap}\ (\mathit{ap}\ (\mathit{Abs}\ (\lambda f \rightarrow \mathit{Abs}\ (\lambda l \rightarrow ...)))\ \mathit{id})\ \mathit{nil}
\end{aligned}
$$

Intermediate closures (example)

$$
\begin{aligned}
&[\![\mathit{map}\ \mathit{id}\ \mathit{nil}]\!] \\
&\qquad = \mathit{ap}\ (\mathit{ap}\ \mathit{map}\ \mathit{id})\ \mathit{nil} \\
&\qquad = \mathit{ap}\ (\mathit{ap}\ (\mathit{Abs}\ (\lambda f \rightarrow \mathit{Abs}\ (\lambda l \rightarrow \ldots)))\ \mathit{id})\ \mathit{nil} \\
&\qquad \rightarrow_\beta\ \ \mathit{ap}\ (\mathit{Abs}\ (\lambda l \rightarrow \ldots))\ \mathit{nil}
\end{aligned}
$$

Intermediate closures (example)

$$
\begin{aligned}
&[\![\mathit{map}\ \mathit{id}\ \mathit{nil}]\!] \\
&\qquad = \mathit{ap}\ (\mathit{ap}\ \mathit{map}\ \mathit{id})\ \mathit{nil} \\
&\qquad = \mathit{ap}\ (\mathit{ap}\ (\mathit{Abs}\ (\lambda f \to \mathit{Abs}\ (\lambda l \to ...)))\ \mathit{id})\ \mathit{nil} \\
&\qquad \to_\beta\ \ \mathit{ap}\ (\mathit{Abs}\ (\lambda l \to ...))\ \mathit{nil} \\
&\qquad\ \ \mathit{nil}
\end{aligned}
$$

Supernumerary arguments (example)

$$\underline{(\lambda x.\ (\lambda y.\ x))\ (\lambda z.\ z)\ 1\ 2}$$

Uncurrying

$$
\begin{aligned}
[\![\underline{x}]\!]\ n &= \hat{x} && \text{if } x < n \\
[\![\underline{x}]\!]\ n &= Con\ \underline{x} && \text{otherwise} \\
[\![\underline{\lambda.\ \cdots \lambda.\ t}]\!]\ n &= Abs_m\ (\lambda \hat{n}\ \ldots\ \widehat{n+m} \rightarrow [\![t]\!]\ (n+m)) \\
[\![\underline{t_1\ \ldots\ t_m}]\!]\ n &= ap_m\ ([\![t_1]\!]\ n) \ldots ([\![t_m]\!]\ n)
\end{aligned}
$$

A family of *ap* operators

1. $ap_n\ (Abs_m\ f)\ t_1\ \ldots\ t_n =$ $Abs_{m-n}\ (f\ t_1\ \ldots\ t_n)$
2. $ap_n\ (Abs_m\ f)\ t_1\ \ldots\ t_n = f\ t_1\ \ldots\ t_n$
3. $ap_n\ (Abs_m\ f)\ t_1\ \ldots\ t_n =$ $ap_{n-m}\ (f\ t_1\ \ldots\ t_m)\ t_{m+1}\ \ldots\ t_n$

Condition on (1): $n < m$
Condition on (2): $n = m$
Condition on (3): $n > m$.

Intermediate closures (revised example)

$$
\begin{aligned}
&[\![map\ id\ nil]\!] \\
&\qquad = ap_2\ map\ id\ nil
\end{aligned}
$$

Intermediate closures (revised example)

$$
\begin{aligned}
&[\![map\ id\ nil]\!] \\
&\qquad = ap_2\ map\ id\ nil \\
&\qquad = ap_2\ (Abs_2\ (\lambda f\ \ l \rightarrow ...))\ id\ nil
\end{aligned}
$$

Intermediate closures (revised example)

$$
\begin{aligned}
&[\![map\ id\ nil]\!] \\
&\qquad = ap_2\ map\ id\ nil \\
&\qquad = ap_2\ (Abs_2\ (\lambda f\ \ l \rightarrow ...))\ id\ nil \\
&\qquad\ \ nil
\end{aligned}
$$

Uncurrying: Remarks

- Drastically reduces number of intermediate closures constructed in the common case.
- No help in pathological cases. But they are rare.
- Only need (small) finite number of *ap* operators.

Embedding pattern matching

Many runtime environments compile pattern matching problems to efficient backtracking automata or decision trees.

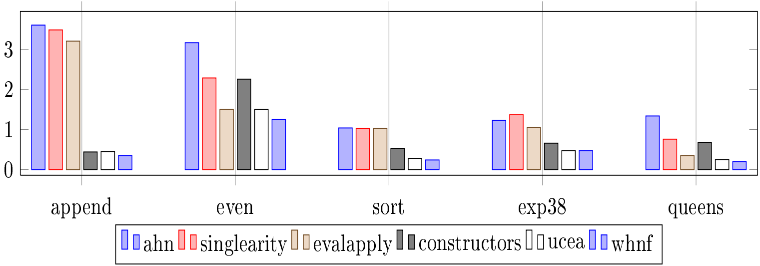
Idea: Extend model with constructors for all constructors in all datatypes introduced in the object language.

Embedding pattern matching

Many runtime environments compile pattern matching problems to efficient backtracking automata or decision trees.

Idea: Extend model with constructors for all constructors in all datatypes introduced in the object language.
$\longrightarrow$ Space and time efficient representation of data.

Microbenchmarks



flavor	append	%	even	%	sort	%	exp3-8	%	queens	%
ahn	3.61	1031	3.17	253	1.04	433	1.23	261	1.34	670
evalapply	3.21	917	1.50	120	1.03	429	1.05	223	0.35	175
singlearity	3.49	997	2.29	183	1.03	429	1.37	191	0.76	380
constructors	0.44	125	2.26	180	0.53	220	0.66	140	0.68	340
ucea	0.45	128	1.50	120	0.28	116	0.47	100	0.25	120
whnf	0.35	100	1.25	100	0.24	100	0.47	100	0.20	100

Macrobenchmarks

variables	2	%	3	%	4	%	5	%
no conv	0.68	94	1.40	93	2.25	77	3.92	3.11
nbe	0.70	97	1.42	94	2.30	79	27.27	20.02
Coq VM	0.72	100	1.50	100	2.92	100	136.2	100

Table: Solving formulae of *n* variables with Cooper's quantifier elimination.

fast cheap general

fast ✓ cheap general

fast ✓ cheap ✓ general

fast ✓ cheap ✓ general ✓

Related work

	fast	cheap	general
Isabelle NbE		✓	✓
TDPE	✓	✓	
Coq VM		✓	✓

Isabelle NbE: compilation to SML (Aehlig et al 2008)

TDPE: type directed partial evaluation (Danvy 1998)

Coq VM: extended version of OCaml virtual machine (Grégoire and Leroy 2002)

Final words

Limitations:

- Fixed evaluation order
- Potential impedance mismatch between object-level pattern matching and meta-level pattern matching.

Future work:

- Short-circuit evaluation.