Paying for a "free" theorem

Matthias Blume
Google

(motivated by joint work with Amal Ahmed)

The (original) goal: fully abstract CPS conversion

The (original) goal: fully abstract CPS conversion

... for System F ...

The (original) goal: fully abstract CPS conversion

... for System F ...
... with recursive types.

Well-known counterexamples due to call/cc-like patterns

$A = \lambda(f{:}\mathrm{int}\to\mathrm{int}, g{:}\mathrm{int}\to\mathrm{int}).(f\ 0;\ g\ 0;\ 0)$
$B = \lambda(f{:}\mathrm{int}\to\mathrm{int}, g{:}\mathrm{int}\to\mathrm{int}).(g\ 0;\ f\ 0;\ 0)$

Well-known counterexamples due to call/cc-like patterns

$A = \lambda(f{:}\mathrm{int}\to\mathrm{int}, g{:}\mathrm{int}\to\mathrm{int}).(f\ 0;\ g\ 0;\ 0)$
$B = \lambda(f{:}\mathrm{int}\to\mathrm{int}, g{:}\mathrm{int}\to\mathrm{int}).(g\ 0;\ f\ 0;\ 0)$

$A' = \lambda(f,g,k{:}\mathrm{int}\to\mathrm{ans}).(f(0,\lambda\_.g(0,\lambda\_.k\ 0)))$
$B' = \lambda(f,g,k{:}\mathrm{int}\to\mathrm{ans}).(g(0,\lambda\_.f(0,\lambda\_.k\ 0)))$

Well-known counterexamples due to call/cc-like patterns

$A = \lambda(f{:}int \to int, g{:}int \to int).(f\ 0;\ g\ 0;\ 0)$
$B = \lambda(f{:}int \to int, g{:}int \to int).(g\ 0;\ f\ 0;\ 0)$

$A' = \lambda(f,g,k{:}int \to \mathit{ans}).(f(0,\lambda\_.g(0,\lambda\_.k\ 0)))$
$B' = \lambda(f,g,k{:}int \to \mathit{ans}).(g(0,\lambda\_.f(0,\lambda\_.k\ 0)))$

$C = \lambda k{:}int \to \mathit{ans}.\ [\cdot]\,(\lambda(\_,\_).k\ 1,\ \lambda(\_,\_).k\ 2,\ k)$

The (original) goal: fully abstract CPS conversion

$e ::= x \mid c \mid \lambda[X\ldots](x{:}T\ldots).e \mid e[T\ldots](e\ldots) \mid \mathrm{fold}[\mu X.T]e \mid \mathrm{unfold}\ e$

$T ::= X \mid \mathrm{int} \mid \forall[X\ldots](T\ldots) \to T \mid \mu X.T$

$v ::= c \mid \lambda[X\ldots](x{:}T\ldots).e \mid \mathrm{fold}[\mu X.T]v$

$E ::= [\cdot] \mid E_0[T\ldots](e\ldots) \mid v[T\ldots](v\ldots E e\ldots)$

$\Gamma ::= x{:}T\ldots$

$\Delta ::= X\ldots$

$\Delta;\Gamma \vdash e : T$

The type translation

$X^+ = X$

$\text{int}^+ = \text{int}$

$(\mu X.T)^+ = \mu X.(T^+)$

The type translation

$X^{+} = X$

$\text{int}^{+} = \text{int}$

$(\mu X.T)^{+} = \mu X.(T^{+})$

$(\forall[X\ldots](S\ldots) \rightarrow T)^{+} = \forall[X\ldots,Y](S^{+}\ldots,T^{+} \rightarrow Y) \rightarrow Y$

CPS translation

$$\Delta;\Gamma \vdash e : T \rightsquigarrow e^*$$

such that

$$\Delta;\Gamma^+ \vdash e^* : T^*$$

where $T^* = \forall\ X\ .\ T^+ \to X$ for some fresh X.

Original goal:

Show that:

if $\Delta;\Gamma \vdash e_1 \approx e_2 : T$

then $\Delta;\Gamma^+ \vdash e_1^* \approx e_2^* : T^*$

A crucial lemma

Suppose $\vdash T$ and $\vdash f : \forall[X](x{:}T, k{:}T \to X) \to X$.

Then for any well-typed choice of a type S and a value v we get:

$$f[S]v \approx v(f[T]\mathit{id}_T)$$

where id_T is the identity for T and $\approx$ is contextual equivalence.

A crucial lemma

Suppose $\vdash T$ and $\vdash f : \forall[X](x{:}T, k{:}T \rightarrow X) \rightarrow X$.

Then for any well-typed choice of a type S and a value v we get:

$$f[S]v \approx v(f[T]id_T)$$

where id_T is the identity for T and ≈ is contextual equivalence.

Thank you, Phil!

Actually...

Actually...

Thanks, but no thanks! ☹

Actually...

Thanks, but no thanks! ☹

The lemma is *not true* in the presence of recursive types, since they introduce non-termination effects. (Other effects would also render it untrue.)

$$f[S]v \approx v(f[T]\mathit{id}_T)$$

$$f[S]v \approx v(f[T]\mathit{id}_T)$$

$T = S = \text{int}$

$f \equiv \lambda[X](k\colon \text{int} \to X).((\lambda(x{:}X).(k\ 1))\ (k\ 0))$

$v \equiv \lambda(x{:}\text{int})\ .\ \text{if } x = 0 \text{ then } \Omega \text{ else } 0$

$$f[S]v \approx v(f[T]\mathit{id}_T)$$

$T = S = \text{int}$

$f \equiv \lambda[X](k\!: \text{int} \to X).((\lambda(x{:}X).(k\ 1))\ (k\ 0))$

$f \equiv \lambda[X](k\!: \text{int} \to X).\ \text{let}\ \_ = k\ 0\ \text{in}\ k\ 1$

$v \equiv \lambda(x{:}\text{int})\ .\ \text{if}\ x = 0\ \text{then}\ \Omega\ \text{else}\ 0$

$$f[S]v \approx v(f[T]\,id_T)$$

T = S = int

f ≡ λ[X](k: int→X).((λ (x:X).(k 1)) (k 0))

f ≡ λ[X](k: int→X). let _ = k 0 in k 1

v ≡ λ(x:int) . if x = 0 then Ω else 0

Assuming a CPS restriction

$T ::= X \mid 1 \mid T \rightarrow T \mid \forall X.T \mid \mu X.T$

$v ::= x \mid () \mid \lambda X{:}T.e \mid \Lambda X.e \mid \mathrm{fold}[\mu X.T]v$

$e ::= v \mid e\ v \mid e[T] \mid \mathrm{let}\ x = \mathrm{unfold}\ v\ \mathrm{in}\ e$

...

kinding context Δ, typing context Γ
small-step semantics with explicit step counts
step-indexed logical relation
sound and complete wrt. contextual equivalence

Claim

For the CPS-restricted System F,
Wadler's "free" theorem holds...

Claim

For the CPS-restricted System F,
Wadler's "free" theorem holds...

... but it is not free

$$f[S]v \approx v(f[T]\mathit{id}_T)$$

$$f[S]v \approx v(f[T]\mathit{id}_T)$$

$$f[S]v \approx v(f[T]\mathit{id}_T)$$

If $f[T]\mathit{id}_T \twoheadrightarrow^* w$ then

$$f[S]v \approx v\ w,$$

and otherwise both $f[T]\mathit{id}_T$ and $f[S]v$ diverge.

Lemma 1: *"If the result of a function is abstract, then the function cannot be invoked unless its result is the final result."*

Lemma 1: *"If the result of a function is abstract, then the function cannot be invoked unless its result is the final result."*

Let $B \neq X$, and $X;f:A \rightarrow X \vdash e : B$.
Then $\exists\ v_P$ such that $X;f:A \rightarrow X \vdash v_P : B$.
Moreover, for any number of steps i, if for some T_0 and some F_0 with $F_0 : A \rightarrow T_0$ it is the case that

$$[X \mapsto T_0, f \mapsto F_0]e \Rightarrow^i v_0,$$

then $v_0 \equiv [X \mapsto T_0, f \mapsto F_0]v_P$. Moreover, in this case for any other T and F such that $F : A \rightarrow T$ it is also true that $[X \mapsto T, f \mapsto F]e \Rightarrow^i [X \mapsto T, f \mapsto F]v_P$.

Proof:

Proof:

By induction on i and case analysis on e.
Notation $\zeta = [X \mapsto T, f \mapsto F]$ and $\zeta_0 = [X \mapsto T_0, f \mapsto F_0]$.

Proof:

By induction on i and case analysis on e.
Notation $\zeta = [X \mapsto T, f \mapsto F]$ and $\zeta_0 = [X \mapsto T_0, f \mapsto F_0]$.

case $e \equiv v_P$: immediate.

Proof:

By induction on i and case analysis on e.
Notation $\zeta = [X \mapsto T, f \mapsto F]$ and $\zeta_0 = [X \mapsto T_0, f \mapsto F_0]$.

case $e \equiv v_P$: immediate.
case $e \equiv e'\ w$:
Have $X;f:A \to X \vdash w : C$ and $X;f:A \to X \vdash e' : C \to B$.
By IH $\exists$ v'_P such that $\zeta_0 e' \Rightarrow^j \zeta_0 v'_P$ and $\zeta e' \Rightarrow^j \zeta v'_P$
with $0 \leq j < i$.
By canonical forms, v'P can be either a variable or some λx:C.
b.

Proof:

By induction on i and case analysis on e.
Notation $\zeta = [X \mapsto T, f \mapsto F]$ and $\zeta_0 = [X \mapsto T_0, f \mapsto F_0]$.

case $e \equiv v_P$: immediate.
case $e \equiv e'\ w$:
Have $X; f{:}A \rightarrow X \vdash w : C$ and $X; f{:}A \rightarrow X \vdash e' : C \rightarrow B$.
By IH $\exists\ v'_P$ such that $\zeta_0 e' \Rightarrow^j \zeta_0 v'_P$ and $\zeta e' \Rightarrow^j \zeta v'_P$
with $0 \leq j < i$.
By canonical forms, v'P can be either a variable or some $\lambda x{:}C.$
b. But the only variable is f, and $B \neq X$.

Proof:

By induction on i and case analysis on e.
Notation $\zeta = [X \mapsto T, f \mapsto F]$ and $\zeta_0 = [X \mapsto T_0, f \mapsto F_0]$.

case $e \equiv v_P$: immediate.
case $e \equiv e'\ w$:
Have $X; f{:}A \to X \vdash w : C$ and $X; f{:}A \to X \vdash e' : C \to B$.
By IH $\exists\ v'_P$ such that $\zeta_0 e' \twoheadrightarrow^j \zeta_0 v'_P$ and $\zeta e' \twoheadrightarrow^j \zeta v'_P$
with $0 \leq j < i$.
By canonical forms, v'P can be either a variable or some λx:C.
b. But the only variable is f, and B ≠ X.
Remainder follows from performing a single reduction step on the redex and re-apply the IH.
case ...

Corollary 3: *If the abstract result type of a function is not part of the overall type, then the computation cannot depend on the function (and the function can, thus, not be invoked at all).*

Corollary 3: *If the abstract result type of a function is not part of the overall type, then the computation cannot depend on the function (and the function can, thus, not be invoked at all).*

Let $X;f:A \to X \vdash e : B$ where $\vdash A$ and $\vdash B$.
Let there be two substitutions $\kappa_1 = [X \mapsto T_1, f \mapsto F_1]$ and $\kappa_2 = [X \mapsto T_2, f \mapsto F_2]$ with $\vdash T_1$, $\vdash T_2$, $\vdash F_1 : A \to T_1$, and $\vdash F_2 : A \to T_2$.
Then $\kappa_1\, e \approx \kappa_2\, e : B$.

Lemma 4: *Wadler's original "free" theorem holds.*

Lemma 4: *Wadler's original "free" theorem holds.*

If $\mathrm{f[T]}\mathit{id}_{\mathrm{T}} \Rightarrow^{*} \mathrm{w}$ then

$$\mathrm{f[S]v} \approx \mathrm{v\ w},$$

and otherwise both $\mathrm{f[T]}\mathit{id}_{\mathrm{T}}$ and f[S]v diverge.