Well-typed programs can't be blamed

Philip Wadler

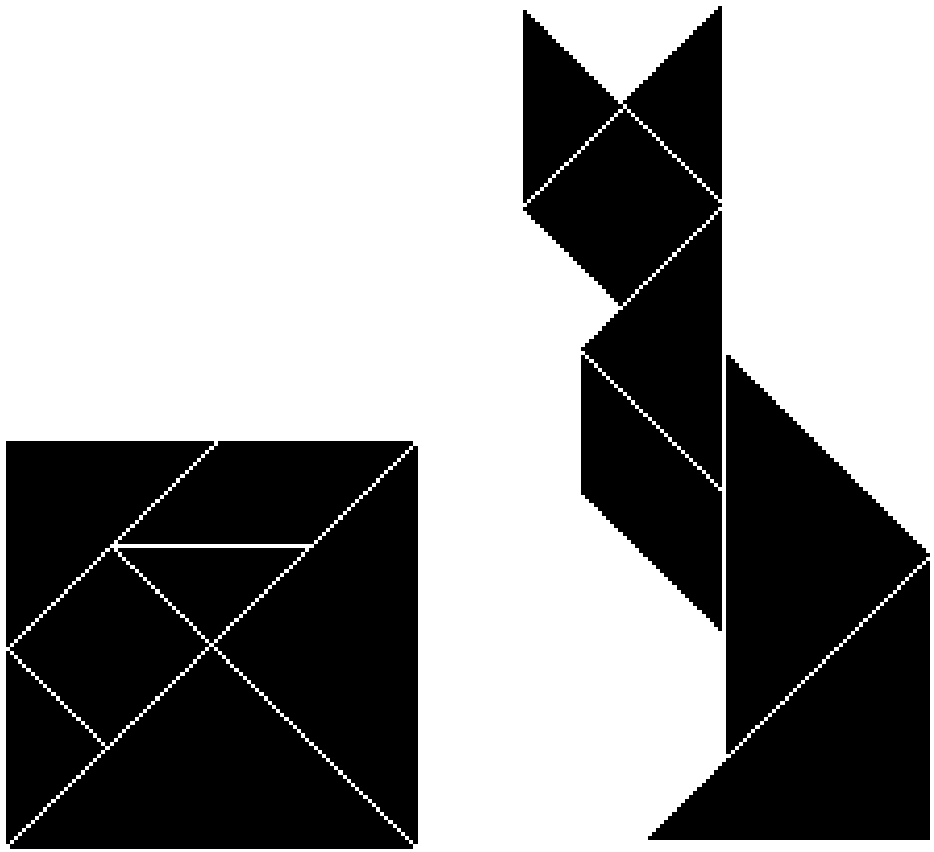
University of Edinburgh

NII Shonan Meeting

26–30 May 2014

Publications

- Wadler and Findler
  Well-typed programs can't be blamed
  ESOP 2009
- Siek and Wadler
  Threesomes, with and without blame
  POPL 2010
- Ahmed, Findler, Siek, and Wadler
  Blame for all
  POPL 2011
- Siek, Thiemann, Wadler
  Blame, coercions, and threesomes, precisely
  draft



A repeated theme

Henglein (1994):

Coercions

Findler and Felleisen (2002):

Contracts

Flanagan (2006):

Hybrid types

Siek and Taha (2006):

Gradual types

A repeated theme

Dynamic in .Net
C#, Visual Basic

JavaScript
TypeScript

Dart

Perl 6.0

Python
Reticulated Python

Part I

From untyped to typed

An untyped program

$$
\begin{array}{l}
\lceil \texttt{let} \\
\quad x = 2 \\
\quad f = \lambda y.\, y + 1 \\
\quad h = \lambda g.\, g\,(g\,x) \\
\texttt{in} \\
\quad h\,f \rceil \\
\longrightarrow \\
\lceil 4 \rceil
\end{array}
$$

A typed program

$$
\begin{array}{l}
\texttt{let} \\
\quad x = 2 \\
\quad f = \lambda y : \texttt{Int}.\, y + 1 \\
\quad h = \lambda g : \texttt{Int} \to \texttt{Int}.\, g\,(g\,x) \\
\texttt{in} \\
\quad h\,f \\
\longrightarrow \\
\quad 4 : \texttt{Int}
\end{array}
$$

A partly typed program—narrowing

$$
\begin{array}{l}
\texttt{let} \\
\quad x = 2 \\
\quad f = \lceil \lambda y.\, y + 1 \rceil : \star \overset{p}{\Rightarrow} \texttt{Int} \to \texttt{Int} \\
\quad h = \lambda g : \texttt{Int} \to \texttt{Int}.\, g\,(g\,x) \\
\texttt{in} \\
\quad h\,f \\
\longrightarrow \\
\quad 4 : \texttt{Int}
\end{array}
$$

A partly typed program—narrowing

$$
\begin{array}{l}
\texttt{let} \\
\quad x = 2 \\
\quad f = \lceil \lambda y.\, \texttt{false} \rceil : \star \overset{p}{\Rightarrow} \texttt{Int} \to \texttt{Int} \\
\quad h = \lambda g : \texttt{Int} \to \texttt{Int}.\, g\,(g\,x) \\
\texttt{in} \\
\quad h\,f \\
\longrightarrow \\
\quad \texttt{blame}\, p
\end{array}
$$

Positive (covariant): blame the term contained in the cast

Another partly typed program—widening

$$
\begin{array}{l}
\texttt{let} \\
\quad x = \lceil 2 \rceil \\
\quad f = (\lambda y : \texttt{Int}.\, y + 1) : \texttt{Int} \to \texttt{Int} \stackrel{p}{\Rightarrow} \star \\
\quad h = \lceil \lambda g.\, g\,(g\,x) \rceil \\
\texttt{in} \\
\quad \lceil h\, f \rceil \\
\longrightarrow \\
\quad \lceil 4 \rceil
\end{array}
$$

Another partly typed program—widening

$$
\begin{array}{l}
\texttt{let} \\
\quad x = \lceil \texttt{true} \rceil \\
\quad f = (\lambda y : \texttt{Int}.\, y + 1) : \texttt{Int} \to \texttt{Int} \overset{p}{\Rightarrow} \star \\
\quad h = \lceil \lambda g.\, g\,(g\,x) \rceil \\
\texttt{in} \\
\quad \lceil h\, f \rceil \\
\longrightarrow \\
\quad \texttt{blame}\, \bar{p}
\end{array}
$$

Negative (contravariant): blame the context containg the cast

Part II

Untyped and supertyped

Untyped = Uni-typed

$$
\begin{array}{lcll}
\lceil x \rceil & = & x & \\
\lceil k \rceil & = & k : A \overset{p}{\Rightarrow} \star & \text{if } \mathrm{ty}(k) = A \\
\lceil op(\vec{M}) \rceil & = & op(\lceil \vec{M} \rceil : \vec{\star} \overset{\vec{p}}{\Rightarrow} \vec{A}) : B \overset{p}{\Rightarrow} \star & \text{if } \mathrm{ty}(op) = \vec{A} \to B \\
\lceil \lambda x.\, N \rceil & = & (\lambda x : \star.\, \lceil N \rceil) : \star \to \star \Rightarrow \star & \\
\lceil L\ M \rceil & = & (\lceil L \rceil : \star \overset{p}{\Rightarrow} \star \to \star)\ \lceil M \rceil &
\end{array}
$$

(slogan due to Dana Scott, repeated by Bob Harper)

Contracts

$$\texttt{Nat} = \{x : \texttt{Int} \mid x \geq 0\}$$

$$\begin{array}{l}
\texttt{let} \\
\quad x = 2 : \texttt{Int} \overset{p}{\Rightarrow} \texttt{Nat} \\
\quad f = (\lambda y : \texttt{Int}.\, y + 1) : \texttt{Int} \to \texttt{Int} \overset{q}{\Rightarrow} \texttt{Nat} \to \texttt{Nat} \\
\quad h = \lambda g : \texttt{Nat} \to \texttt{Nat}.\, g\,(g\,x) \\
\texttt{in} \\
\quad h\,f \\
\longrightarrow \\
\quad 4 : \texttt{Nat}
\end{array}$$

Part III

The Blame Calculus

Notation

It took us four years to find the right notation!

$$\langle A \Rightarrow B \rangle^p \; s$$

$$\langle B \Leftarrow A \rangle^p \; s$$

$$s : A \stackrel{p}{\Rightarrow} B$$

We want composition to be easy to read:

$$\langle B \Rightarrow C \rangle^q \; \langle A \Rightarrow B \rangle^p \; s$$

$$\langle C \Leftarrow B \rangle^q \; \langle B \Leftarrow A \rangle^p \; s$$

$$s : A \stackrel{p}{\Rightarrow} B : B \stackrel{q}{\Rightarrow} C$$

And there is a convenient abbreviation:

$$s : A \stackrel{p}{\Rightarrow} B \stackrel{q}{\Rightarrow} C$$

Blame calculus: Compatibility

$$\frac{\Gamma \vdash M : A \qquad A \prec B}{\Gamma \vdash (M : A \overset{p}{\Rightarrow} B) : B}$$

$$\frac{}{A \prec \star} \qquad \frac{}{\star \prec A}$$

$$\frac{}{\iota \prec \iota}$$

$$\frac{A' \prec A \qquad B \prec B'}{A \to B \prec A' \to B'}$$

Reductions

Ground types $G ::= \iota \mid \star \to \star$

$$(V : A \to B \overset{p}{\Rightarrow} A' \to B')\, W \longrightarrow (V\, (W : A' \overset{\bar{p}}{\Rightarrow} A) : B \overset{p}{\Rightarrow} B')$$

$$V : \iota \overset{p}{\Rightarrow} \iota \longrightarrow V$$

$$V : A \overset{p}{\Rightarrow} \star \longrightarrow V : A \overset{p}{\Rightarrow} G \Rightarrow \star \quad \text{if } \star \neq A \prec G$$

$$V : G \Rightarrow \star \overset{p}{\Rightarrow} A \longrightarrow \begin{cases} V : G \overset{p}{\Rightarrow} A & \text{if } G \prec A \\ \texttt{blame}\ p & \text{if } G \not\prec A \end{cases}$$

Blame

$$\lceil 2 \rceil : \star \overset{p}{\Rightarrow} \texttt{Int}$$

$$=$$

$$2 : \texttt{Int} \Rightarrow \star \overset{p}{\Rightarrow} \texttt{Int}$$

$$\longrightarrow$$

$$2$$

$$\lceil \texttt{true} \rceil : \star \overset{p}{\Rightarrow} \texttt{Int}$$

$$=$$

$$\texttt{true} : \texttt{Bool} \Rightarrow \star \overset{p}{\Rightarrow} \texttt{Int}$$

$$\longrightarrow$$

$$\texttt{blame}\, p$$

The Blame Game—widening

$$((\lambda y : \texttt{Int}.\, y + 1) : \texttt{Int} \to \texttt{Int} \overset{p}{\Rightarrow} \star \to \star)\, \lceil 2 \rceil$$

$$\longrightarrow$$

$$(\lambda y : \texttt{Int}.\, y + 1)\, (\lceil 2 \rceil : \star \overset{\bar{p}}{\Rightarrow} \texttt{Int}) : \texttt{Int} \overset{p}{\Rightarrow} \star$$

$$\longrightarrow$$

$$\lceil 3 \rceil$$

The Blame Game—widening

$$((\lambda y : \mathtt{Int}.\, y + 1) : \mathtt{Int} \to \mathtt{Int} \overset{p}{\Rightarrow} \star \to \star)\, \lceil \mathtt{true} \rceil$$

$$\longrightarrow$$

$$(\lambda y : \mathtt{Int}.\, y + 1)\, (\lceil \mathtt{true} \rceil : \star \overset{\overline{p}}{\Rightarrow} \mathtt{Int}) : \mathtt{Int} \overset{p}{\Rightarrow} \star$$

$$\longrightarrow$$

$$\mathtt{blame}\, \overline{p}$$

Widening can give rise to negative blame, but never positive blame

The Blame Game—narrowing

$$((\lambda y : \star.\, \lceil y + 1 \rceil) : \star \to \star \overset{p}{\Rightarrow} \mathtt{Int} \to \mathtt{Int})\, 2$$

$$\longrightarrow$$

$$(\lambda y : \star.\, \lceil y + 1 \rceil)\, (2 : \mathtt{Int} \overset{\bar{p}}{\Rightarrow} \star) : \star \overset{p}{\Rightarrow} \mathtt{Int}$$

$$\longrightarrow$$

$$3$$

The Blame Game—narrowing

$$(\lambda y : \star.\lceil \texttt{false} \rceil) : \star \to \star \stackrel{p}{\Rightarrow} \texttt{Int} \to \texttt{Int})\ \texttt{2}$$

$$\longrightarrow$$

$$(\lambda y : \star.\lceil \texttt{false} \rceil)\,(\texttt{2} : \texttt{Int} \stackrel{\bar{p}}{\Rightarrow} \star) : \star \stackrel{p}{\Rightarrow} \texttt{Int}$$

$$\longrightarrow$$

$$\texttt{blame}\ p$$

Narrowing can give rise to positive blame, but never negative blame

Part IV

Subtyping

$$<: \qquad <:^{+} \qquad <:^{-} \qquad <:_{n}$$

Subtype

$$\frac{}{\star <: \star}$$

$$\frac{}{\iota <: \iota} \qquad \frac{A <: G}{A <: \star}$$

$$\frac{A' <: A \qquad B <: B'}{A \to B <: A' \to B'}$$

Example:

$$\frac{\dfrac{\dfrac{}{\texttt{Int} <: \texttt{Int}}}{\texttt{Int} <: \star} \qquad \dfrac{\dfrac{}{\texttt{Int} <: \texttt{Int}}}{\texttt{Int} <: \star}}{\star \to \texttt{Int} <: \texttt{Int} \to \star}$$

Positive subtype—widening

$$\frac{}{A <:^{+} \star}$$

$$\frac{}{\iota <: \iota}$$

$$\frac{A' <:^{-} A \qquad B <:^{+} B'}{A \to B <:^{+} A' \to B'}$$

Example:

$$\frac{\dfrac{}{\star <:^{-} \texttt{Int}} \qquad \dfrac{}{\texttt{Int} <:^{+} \star}}{\texttt{Int} \to \texttt{Int} <: \star \to \star}$$

Negative subtype—narrowing

$$\frac{}{\star <:^{-} A}$$

$$\frac{}{\iota <: \iota} \qquad \frac{A <:^{-} G}{A <:^{-} \star}$$

$$\frac{A' <:^{+} A \qquad B <:^{-} B'}{A \to B <:^{-} A' \to B'}$$

Example:

$$\frac{\dfrac{}{\texttt{Int} <:^{+} \star} \qquad \dfrac{}{\star <:^{-} \texttt{Int}}}{\star \to \star <:^{-} \texttt{Int} \to \texttt{Int}}$$

Naive subtype

$$\frac{}{A <:_n \star}$$

$$\frac{}{\iota <:_n \iota}$$

$$\frac{A <:_n A' \qquad B <:_n B'}{A \to B <:_n A' \to B'}$$

Example:

$$\frac{\dfrac{}{\texttt{Int} <:_n \star} \qquad \dfrac{}{\texttt{Int} <:_n \star}}{\texttt{Int} \to \texttt{Int} <:_n \star \to \star}$$

Part V

The Blame Theorem

Blame Safety

$$\frac{}{x \texttt{ safe } p} \qquad \frac{N \texttt{ safe } p}{\lambda x.\, N \texttt{ safe } p} \qquad \frac{L \texttt{ safe } p \qquad M \texttt{ safe } p}{L\ M \texttt{ safe } p}$$

$$\frac{M \texttt{ safe } p \qquad A <:^{+} B}{M : A \overset{p}{\Rightarrow} B \texttt{ safe } p}$$

$$\frac{M \texttt{ safe } p \qquad A <:^{-} B}{M : A \overset{\bar{p}}{\Rightarrow} B \texttt{ safe } p}$$

$$\frac{s \texttt{ safe } p \qquad p \neq q \qquad \bar{p} \neq q}{M : A \overset{q}{\Rightarrow} B \texttt{ safe } p}$$

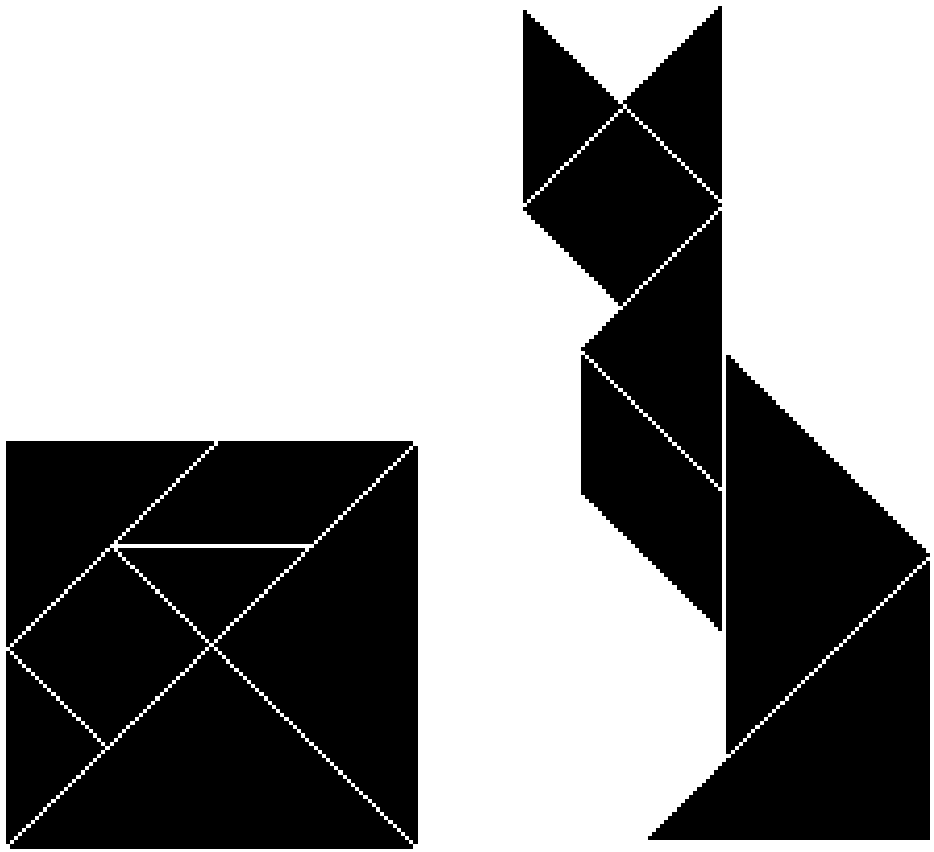
The Blame Theorem

Preservation

If M safe p and $M \longrightarrow N$ then N safe p.

Progress

If M safe p then $M \not\longrightarrow \texttt{blame}\ p$.



The First Tangram Theorem

$$A <: B \text{ if and only if } A <:^{+} B \text{ and } A <:^{-} B$$

The First Blame Corollary

Let M be a term where $N : A \overset{p}{\Rightarrow} B$ is the only subterm with label p.
If $A <: B$ then $M \not\longrightarrow \texttt{blame}\ p$ and $M \not\longrightarrow \texttt{blame}\ \bar{p}$.

The Second Tangram Theorem

$$A <:_n B \text{ if and only if } A <:^+ B \text{ and } B <:^- A$$

The Second Blame Corollary

Let M be a term where $N : A \stackrel{p}{\Rightarrow} B$ is the only subterm with label p.
If $A <:_n B$ then $M \not\longrightarrow \texttt{blame}\ p$.

Let M be a term where $N : A \stackrel{p}{\Rightarrow} B$ is the only subterm with label p.
If $B <:_n A$ then $M \not\longrightarrow \texttt{blame}\ \overline{p}$.

Part VI

Blame and coercions

Lambda calculus with coercions

$$\frac{\Gamma \vdash M : A \qquad c : A \Rightarrow B}{\Gamma \vdash M : A \overset{c}{\Rightarrow} B}$$

Coercion typing

$$\frac{}{\mathtt{id}(A) : A \Rightarrow A}$$

$$\frac{}{(G\,!) : G \Rightarrow \star}$$

$$\frac{}{(p\,?\,G) : \star \Rightarrow G}$$

$$\frac{c : A' \Rightarrow A \quad d : B \Rightarrow B'}{(c \rightarrow d) : A \rightarrow B \Rightarrow A' \rightarrow B'}$$

$$\frac{c : A \Rightarrow B \quad d : B \Rightarrow C}{(c\,;d) : A \Rightarrow C}$$

Blame safety is preserved and reflected

Let cast $A \stackrel{p}{\Rightarrow} B$ translate to coercion c.

$A <:^{+} B$ if and only if p does not appear in c.

$A <:^{-} B$ if and only if $\bar{p}$ does not appear in c.

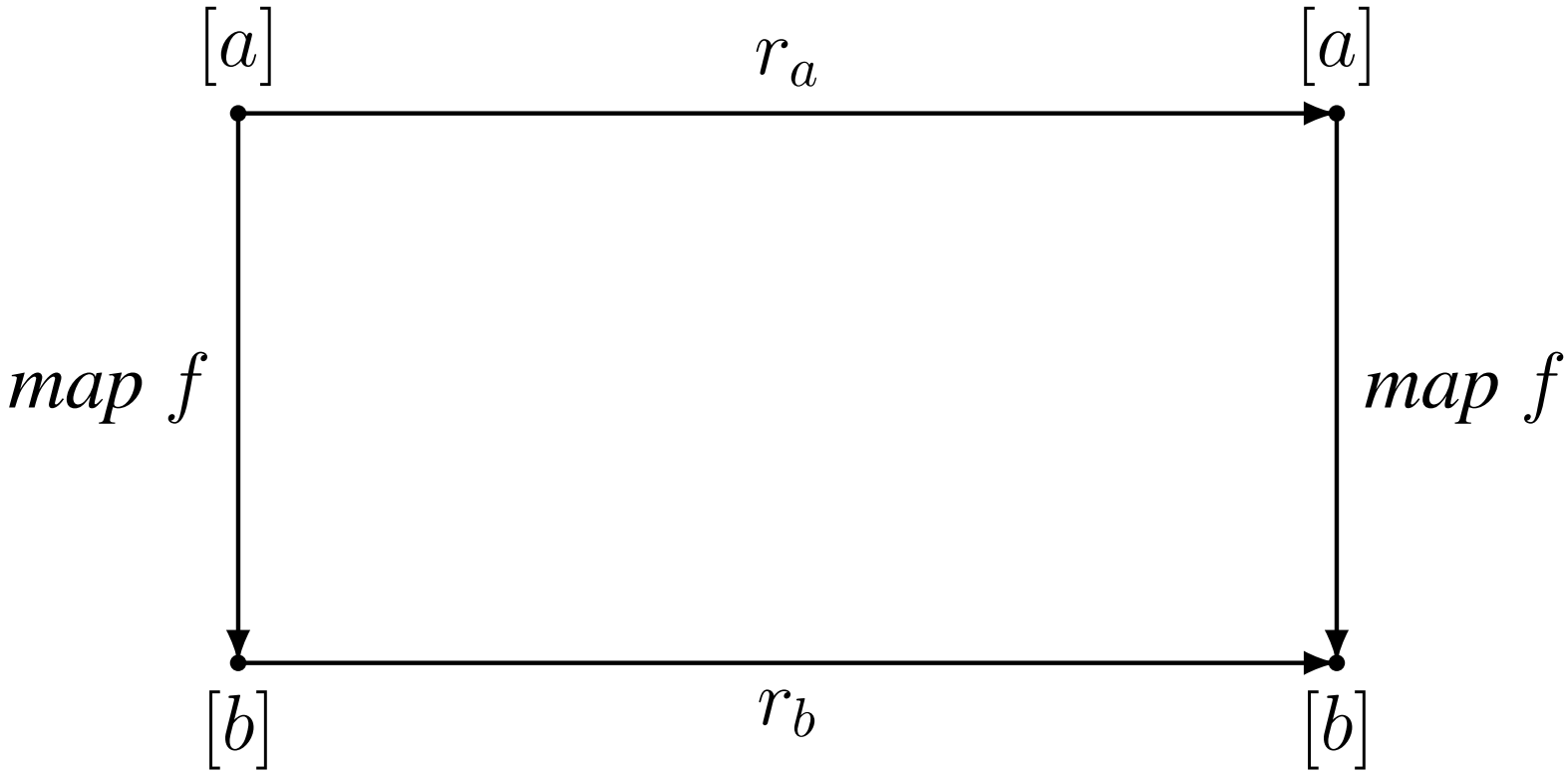
Part VII

Polymorphism

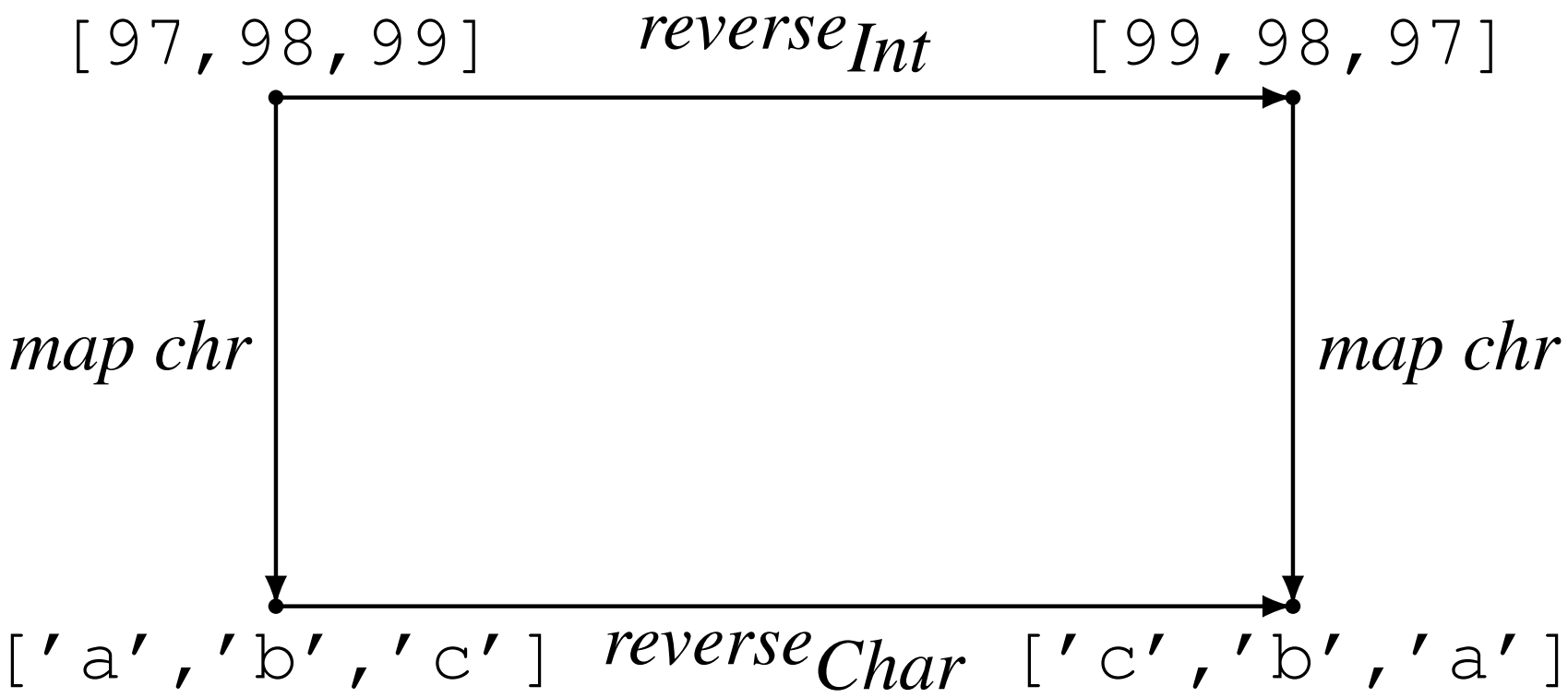
A magic trick

$$r :: [a] \longrightarrow [a]$$

Theorems for Free!



Theorems for Free!



Explicit binding

$$\frac{\Gamma, X{:=}A \vdash N : B \quad X \notin \mathrm{ftv}(B)}{\Gamma \vdash \nu X{:=}A.\, N : B}$$

$$\frac{\Gamma \vdash N : B \quad (X{:=}A) \in \Gamma}{\Gamma \vdash N : B[X{:=}A]} \qquad \frac{\Gamma \vdash N : B[X{:=}A] \quad (X{:=}A) \in \Gamma}{\Gamma \vdash N : B}$$

$$(\Lambda X.\, N)\, A \quad \longrightarrow \quad \nu X{:=}A.\, N$$

Global store vs. Local bindings

George Neis, Derek Dreyer, and Andreas Rossberg. Non-parametric parametricity. ICFP 2009, Edinburgh.

Compatibility and reductions

$$\frac{}{X \prec X} \qquad \frac{A \prec B}{A \prec \forall X.\, B}\; X \notin \mathrm{ftv}(A) \qquad \frac{A[X{:=}\star] \prec B}{\forall X.\, A \prec B}$$

$$\begin{array}{lcll}
V : A \overset{p}{\Rightarrow} (\forall X.\, B) & \longrightarrow & \Lambda X.\, (V : A \overset{p}{\Rightarrow} B) & \text{if } X \notin \mathrm{ftv}(A) \\
V : (\forall X.\, A) \overset{p}{\Rightarrow} B & \longrightarrow & (V\ \star) : A[X{:=}\star] \overset{p}{\Rightarrow} B & \\
V : X \overset{p}{\Rightarrow} \star \overset{q}{\Rightarrow} X & \longrightarrow & V & \\
V : X \overset{p}{\Rightarrow} \star \overset{q}{\Rightarrow} Y & \longrightarrow & \mathtt{blame}\ q & \text{if } X \neq Y
\end{array}$$

Instantiate

$$V : \forall X.\, A \overset{p}{\Rightarrow} B \longrightarrow V \star : A[X{:=}\star] \overset{p}{\Rightarrow} B$$

$$K = \lambda x : X.\, \lambda y : X.\, x$$

$$\begin{aligned}
&(((\Lambda X.\, K) : \forall X.\, X{\to}X{\to}X \overset{p}{\Rightarrow} \star{\to}\star{\to}\star)\ 42\ 7 \\
\longrightarrow\ &(((\Lambda X.\, K)\ \star : \star{\to}\star{\to}\star \overset{p}{\Rightarrow} \star{\to}\star{\to}\star)\ 42\ 7 \\
\longrightarrow\ &\nu X{:=}\star.\, (K : \star{\to}\star{\to}\star \overset{p}{\Rightarrow} \star{\to}\star{\to}\star)\ 42\ 7 \\
\longrightarrow\ &\nu X{:=}\star.\, (K\ (42 : \star \overset{\bar{p}}{\Rightarrow} \star)\ (7 : \star \overset{\bar{p}}{\Rightarrow} \star)) : \star \overset{p}{\Rightarrow} \star \\
\longrightarrow\ &\nu X{:=}\star.\, (42 : \star \overset{\bar{p}}{\Rightarrow} \star) : \star \overset{p}{\Rightarrow} \star \\
\longrightarrow\ &\nu X{:=}\star.\, 42 \\
\longrightarrow\ &42
\end{aligned}$$

$\star$ is a Jack-of-all-Trades

$$V : \forall X.\, A \stackrel{p}{\Rightarrow} B \longrightarrow V\,\star : A[X{:=}\star] \stackrel{p}{\Rightarrow} B$$

$$K = \lambda x : X.\, \lambda y : X.\, x$$

$$\begin{aligned}
&((\Lambda X.\, K) : \forall X.\, X{\to}X{\to}X \stackrel{p}{\Rightarrow} \mathtt{Int}{\to}\mathtt{Int}{\to}\mathtt{Int})\ \mathtt{42}\ \mathtt{7}\\
\longrightarrow\ &((\Lambda X.\, K)\,\star : \star{\to}\star{\to}\star \stackrel{p}{\Rightarrow} \mathtt{Int}{\to}\mathtt{Int}{\to}\mathtt{Int})\ \mathtt{42}\ \mathtt{7}\\
\longrightarrow\ &\nu X{:=}\star.\, (K : \star{\to}\star{\to}\star \stackrel{p}{\Rightarrow} \mathtt{Int}{\to}\mathtt{Int}{\to}\mathtt{Int})\ \mathtt{42}\ \mathtt{7}\\
\longrightarrow\ &\nu X{:=}\star.\, (K\ (\mathtt{42} : \mathtt{Int} \stackrel{\bar{p}}{\Rightarrow} \star)\ (\mathtt{7} : \mathtt{Int} \stackrel{\bar{p}}{\Rightarrow} \star)) : \star \stackrel{p}{\Rightarrow} \mathtt{Int}\\
\longrightarrow\ &\nu X{:=}\star.\, (\mathtt{42} : \mathtt{Int} \stackrel{\bar{p}}{\Rightarrow} \star) : \star \stackrel{p}{\Rightarrow} \mathtt{Int}\\
\longrightarrow\ &\nu X{:=}\star.\, \mathtt{42}\\
\longrightarrow\ &\mathtt{42}
\end{aligned}$$

. . . but master of none

$$V : \forall X.\, A \stackrel{p}{\Rightarrow} B \;\longrightarrow\; V \star : A[X{:=}\star] \stackrel{p}{\Rightarrow} B$$

$$K = \lambda x : X.\, \lambda y : X.\, x$$

$$\begin{aligned}
&((\Lambda X.\, K) : \forall X.\, X{\to}X{\to}X \stackrel{p}{\Rightarrow} \texttt{Int}{\to}\texttt{Bool}{\to}\texttt{Int})\ \texttt{42}\ \texttt{true}\\
\longrightarrow\ &((\Lambda X.\, K)\star : \star{\to}\star{\to}\star \stackrel{p}{\Rightarrow} \texttt{Int}{\to}\texttt{Bool}{\to}\texttt{Int})\ \texttt{42}\ \texttt{true}\\
\longrightarrow\ &\nu X{:=}\star.\, (K : \star{\to}\star{\to}\star \stackrel{p}{\Rightarrow} \texttt{Int}{\to}\texttt{Bool}{\to}\texttt{Int})\ \texttt{42}\ \texttt{true}\\
\longrightarrow\ &\nu X{:=}\star.\, (K\ (\texttt{42} : \texttt{Int} \stackrel{\bar{p}}{\Rightarrow} \star)\ (\texttt{true} : \texttt{Bool} \stackrel{\bar{p}}{\Rightarrow} \star)) : \star \stackrel{p}{\Rightarrow} \texttt{Int}\\
\longrightarrow\ &\nu X{:=}\star.\, (\texttt{42} : \texttt{Int} \stackrel{\bar{p}}{\Rightarrow} \star) : \star \stackrel{p}{\Rightarrow} \texttt{Int}\\
\longrightarrow\ &\nu X{:=}\star.\, \texttt{42}\\
\longrightarrow\ &\texttt{42}
\end{aligned}$$

Generalise

$$V : A \stackrel{p}{\Rightarrow} \forall X.\, B \;\longrightarrow\; \Lambda X.\, (V : A \stackrel{p}{\Rightarrow} B) \qquad X \text{ not free in } A$$

$$K = \lambda x : \star.\, \lambda y : \star.\, x$$

$$\begin{aligned}
&(K : \star \to \star \to \star \stackrel{p}{\Rightarrow} \forall X.\, \forall Y.\, X \to Y \to X)\ \mathtt{Int}\ \mathtt{Int}\ \mathtt{42}\ \mathtt{7} \\
\longrightarrow\ &(\Lambda X.\, \Lambda Y.\, (K : \star \to \star \to \star \stackrel{p}{\Rightarrow} X \to Y \to X))\ \mathtt{Int}\ \mathtt{Int}\ \mathtt{42}\ \mathtt{7} \\
\longrightarrow\ &\nu X{:=}\mathtt{Int}.\, \nu Y{:=}\mathtt{Int}.\, (K : \star \to \star \to \star \stackrel{p}{\Rightarrow} X \to Y \to X)\ \mathtt{42}\ \mathtt{7} \\
\longrightarrow\ &\nu X{:=}\mathtt{Int}.\, \nu Y{:=}\mathtt{Int}.\, (K\ (\mathtt{42} : X \stackrel{\bar{p}}{\Rightarrow} \star)\ (\mathtt{7} : Y \stackrel{\bar{p}}{\Rightarrow} \star)) : \star \stackrel{p}{\Rightarrow} X \\
\longrightarrow\ &\nu X{:=}\mathtt{Int}.\, \nu Y{:=}\mathtt{Int}.\, (\mathtt{42} : X \stackrel{\bar{p}}{\Rightarrow} \star) : \star \stackrel{p}{\Rightarrow} X \\
\longrightarrow\ &\nu X{:=}\mathtt{Int}.\, \nu Y{:=}\mathtt{Int}.\, \mathtt{42} \\
\longrightarrow\ &\mathtt{42}
\end{aligned}$$

Enforcing semantic parametricity

$$V : A \overset{p}{\Rightarrow} \forall X.\, B \;\longrightarrow\; \Lambda X.\, (V : A \overset{p}{\Rightarrow} B) \qquad X \text{ not free in } A$$

$$K' = \lambda x : \star.\, \lambda y : \star.\, y$$

$$\begin{aligned}
&(K' : \star \to \star \to \star \overset{p}{\Rightarrow} \forall X.\, \forall Y.\, X \to Y \to X)\ \texttt{Int}\ \texttt{Int}\ \texttt{42}\ \texttt{7} \\
\longrightarrow\ &(\Lambda X.\, \Lambda Y.\, (K' : \star \to \star \to \star \overset{p}{\Rightarrow} X \to Y \to X))\ \texttt{Int}\ \texttt{Int}\ \texttt{42}\ \texttt{7} \\
\longrightarrow\ &\nu X{:=}\texttt{Int}.\, \nu Y{:=}\texttt{Int}.\, (K' : \star \to \star \to \star \overset{p}{\Rightarrow} X \to Y \to X)\ \texttt{42}\ \texttt{7} \\
\longrightarrow\ &\nu X{:=}\texttt{Int}.\, \nu Y{:=}\texttt{Int}.\, (K'\ (\texttt{42} : X \overset{\overline{p}}{\Rightarrow} \star)\ (\texttt{7} : Y \overset{\overline{p}}{\Rightarrow} \star)) : \star \overset{p}{\Rightarrow} X \\
\longrightarrow\ &\nu X{:=}\texttt{Int}.\, \nu Y{:=}\texttt{Int}.\, (\texttt{7} : Y \overset{\overline{p}}{\Rightarrow} \star) : \star \overset{p}{\Rightarrow} X \\
\longrightarrow\ &\texttt{blame}\ p
\end{aligned}$$

You can't look inside a seal

$$(V : G \overset{p}{\Rightarrow} \star)\ \texttt{is}\ H \longrightarrow \begin{cases} \texttt{true} & \text{if } G = H \\ \texttt{false} & \text{if } G \neq H \text{ and } G \neq X \\ \texttt{blame}\ p_{\texttt{is}} & \text{if } G = X \end{cases}$$

$$\mathit{test} = \lambda x.\, \texttt{if}\ (x\ \texttt{is}\ \texttt{Int})\ \texttt{then}\ \lceil x + 1 \rceil\ \texttt{else}\ x$$

$$(\mathit{test} : \star \to \star \overset{p}{\Rightarrow} \forall X.\, X \to X)\ \texttt{Int}\ 2$$

$$\longrightarrow \nu X{:=}\texttt{Int}.\, \mathit{test}\ (2 : X \overset{\bar{p}}{\Rightarrow} \star) : \star \overset{p}{\Rightarrow} X$$

$$\longrightarrow \nu X{:=}\texttt{Int}.\, \texttt{if}\ ((2 : X \Rightarrow \star)\ \texttt{is}\ \texttt{Int})\ \texttt{then}\ \cdots\ \texttt{else}\ \cdots$$

$$\longrightarrow \texttt{blame}\ p_{\texttt{is}}$$

Part VIII

Threesomes

Even and odd, untyped

$$
\begin{array}{l}
\lceil \texttt{let} \\
\quad \mathit{even}^\star = \lambda n.\, \texttt{if}\ n = 0 \\
\qquad\qquad\qquad \texttt{then true} \\
\qquad\qquad\qquad \texttt{else}\ \mathit{odd}^\star\,(n-1) \\
\quad \mathit{odd}^\star = \lambda n.\, \texttt{if}\ n = 0 \\
\qquad\qquad\qquad\ \texttt{then false} \\
\qquad\qquad\qquad\ \texttt{else}\ \mathit{even}^\star\,(n-1) \\
\texttt{in} \\
\quad \mathit{even}^\star\ 2 \rceil
\end{array}
$$

Even and odd, partly typed

$$
\begin{array}{l}
\texttt{let} \\
\quad even^{\star} = \lceil \lambda n.\, \texttt{if}\ n = 0 \\
\qquad\qquad\qquad\qquad \texttt{then true} \\
\qquad\qquad\qquad\qquad \texttt{else}\ odd^{\star}\ (n-1) \rceil \\
\quad odd = \lambda n : \texttt{Int}.\, \texttt{if}\ n = 0 \\
\qquad\qquad\qquad\qquad\quad \texttt{then false} \\
\qquad\qquad\qquad\qquad\quad \texttt{else}\ even\ (n-1) \\
\quad even = even^{\star} : \star \stackrel{p}{\Rightarrow} \texttt{Int} \rightarrow \texttt{Int} \\
\quad odd^{\star} = odd : \texttt{Int} \rightarrow \texttt{Int} \stackrel{q}{\Rightarrow} \star \\
\texttt{in} \\
\quad \lceil even^{\star}\ 2 \rceil
\end{array}
$$

Goodbye tail recursion!

$$\lceil even^{\star}\ 4 \rceil$$

$$\longrightarrow$$

$$(odd\ 3) : \texttt{Bool} \overset{q}{\Rightarrow} \star$$

$$\longrightarrow$$

$$\lceil even^{\star}\ 2 \rceil : \star \overset{p}{\Rightarrow} \texttt{Bool} \overset{q}{\Rightarrow} \star$$

$$\longrightarrow$$

$$(odd\ 1) : \texttt{Bool} \overset{q}{\Rightarrow} \star \overset{p}{\Rightarrow} \texttt{Bool} \overset{q}{\Rightarrow} \star$$

$$\longrightarrow$$

$$\lceil even^{\star}\ 0 \rceil : \star \overset{p}{\Rightarrow} \texttt{Bool} \overset{q}{\Rightarrow} \star \overset{p}{\Rightarrow} \texttt{Bool} \overset{q}{\Rightarrow} \star$$

$$\longrightarrow$$

$$\lceil \texttt{true} \rceil$$

Recovering tail recursion!

$$
\begin{aligned}
& \lceil even^{\star}\ 4 \rceil \\
\longrightarrow\ & \\
& (odd\ 3) : \texttt{Bool} \overset{q}{\Rightarrow} \star \\
\longrightarrow\ & \\
& \lceil even^{\star}\ 2 \rceil : \star \overset{p}{\Rightarrow} \texttt{Bool} \overset{q}{\Rightarrow} \star \\
\longrightarrow\ & \\
& (odd\ 1) : \texttt{Bool} \overset{q}{\Rightarrow} \star \\
\longrightarrow\ & \\
& \lceil even^{\star}\ 0 \rceil : \star \overset{p}{\Rightarrow} \texttt{Bool} \overset{q}{\Rightarrow} \star \\
\longrightarrow\ & \\
& \lceil \texttt{true} \rceil
\end{aligned}
$$

Part IX

Twosomes: a small change to blame calculus

Add the empty type

$$\text{Types} \quad A, B, C \quad ::= \quad \iota \mid A \to B \mid \star \mid \bot$$

Every type is incompatible with $\bot$.

$$A \not\asymp \bot \qquad \bot \not\asymp A$$

There are no values of type $\bot$.

Allow casts between any types

Change one type rule

$$\frac{\Gamma \vdash M : A \qquad A \prec B}{\Gamma \vdash (M : A \overset{p}{\Rightarrow} B) : B}$$

becomes

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash (M : A \overset{p}{\Rightarrow} B) : B}$$

Add one reduction

$$V : A \overset{p}{\Rightarrow} \bot \longrightarrow \texttt{blame}\, p$$

Naive subtype

$$\frac{}{A <:_n \star}$$

$$\frac{}{\iota <:_n \iota}$$

$$\frac{A <:_n A' \qquad B <:_n B'}{A \to B <:_n A' \to B'}$$

$$\frac{}{\bot <:_n A}$$

Meet

$$
\begin{aligned}
A \mathbin{\&} \star &= A \\
\star \mathbin{\&} A &= A \\
\iota \mathbin{\&} \iota &= \iota \\
(A \to B) \mathbin{\&} (A' \to B') &= (A \mathbin{\&} A') \to (B \mathbin{\&} B') \\
A \mathbin{\&} B &= \bot \qquad \text{otherwise}
\end{aligned}
$$

$A \mathbin{\&} B$ is the greatest lower bound of A and B.

$C <:_n A \mathbin{\&} B$ if and only if $C <:_n A$ and $C <:_n B$

First factoring theorem

$$V : A \stackrel{p}{\Rightarrow} B \quad = \quad V : A \stackrel{p}{\Rightarrow} A \,\&\, B \stackrel{p}{\Rightarrow} B$$

An example

$$V : (\texttt{Int} \to \star) \stackrel{p}{\Rightarrow} (\star \to \texttt{Bool})$$

$$=$$

$$V : (\texttt{Int} \to \star) \stackrel{p}{\Rightarrow} (\texttt{Int} \to \texttt{Bool}) \stackrel{p}{\Rightarrow} (\star \to \texttt{Bool})$$

Second factoring theorem

$$V : A \stackrel{p}{\Rightarrow} B \quad = \quad V : A \stackrel{p}{\Rightarrow} \star \stackrel{p}{\Rightarrow} B$$

An example

$$V : (\texttt{Int} \to \star) \stackrel{p}{\Rightarrow} (\star \to \texttt{Bool})$$

$$=$$

$$V : (\texttt{Int} \to \star) \stackrel{p}{\Rightarrow} \star \stackrel{p}{\Rightarrow} (\star \to \texttt{Bool})$$

Third factoring theorem

$$V : A \stackrel{p}{\Rightarrow} B \quad = \quad V : A \stackrel{p}{\Rightarrow} C \stackrel{p}{\Rightarrow} B$$

whenever $A \mathbin{\&} B <:_n C$

This result subsumes the other two.

Take $C = A \mathbin{\&} B$ for the first theorem.

Take $C = \star$ for the second theorem.

Part X

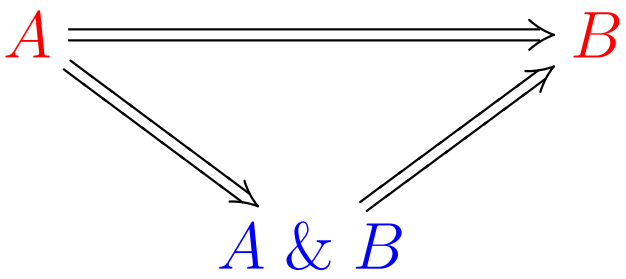
Threesomes, without blame

Threesomes

Let A, B, C, P, Q range over types.

$$V : A \stackrel{P}{\Longrightarrow} B \quad =_{\text{def}} \quad V : A \Rightarrow P \Rightarrow B$$

whenever $P <:_n A$ and $P <:_n B$.



Combining threesomes

$$V : A \overset{P}{\Longrightarrow} B \overset{Q}{\Longrightarrow} C$$

$=$ Def'n: $P <:_n A, P <:_n B, Q <:_n B, Q <:_n C$

$$V : A \Rightarrow P \Rightarrow B \Rightarrow Q \Rightarrow C$$

$=$ 3'rd thm, $P \,\&\, Q <:_n B$

$$V : A \Rightarrow P \Rightarrow Q \Rightarrow C$$

$=$ 1'st thm

$$V : A \Rightarrow P \Rightarrow P \,\&\, Q \Rightarrow Q \Rightarrow C$$

$=$ 3'rd thm, $A \,\&\, P \,\&\, Q <:_n P, P \,\&\, Q \,\&\, C <:_n Q$

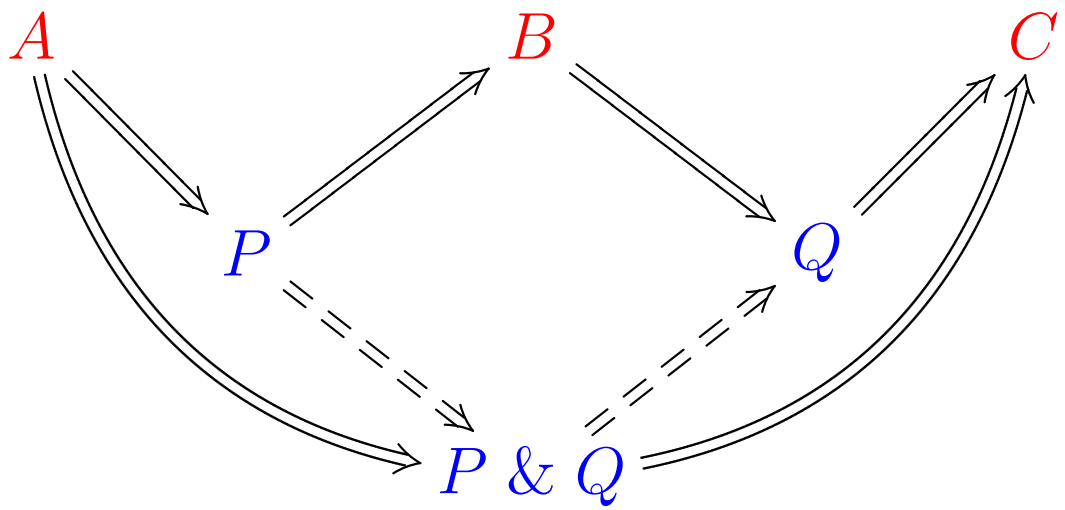
$$V : A \Rightarrow P \,\&\, Q \Rightarrow C$$

$=$ Def'n: $P \,\&\, Q <:_n A, P \,\&\, Q <:_n C$

$$V : A \overset{P\&Q}{\Longrightarrow} C$$

Combining threesomes

$$V : A \overset{P}{\Longrightarrow} B \overset{Q}{\Longrightarrow} C \quad \longrightarrow \quad V : A \overset{P\&Q}{\Longrightarrow} C$$



Recovering tail recursion!

$$
\begin{array}{l}
\lceil even^{\star}\ 4 \rceil \\
\longrightarrow \\
(odd\ 3) : \texttt{Bool} \overset{\texttt{Bool}}{\Longrightarrow} \star \\
\longrightarrow \\
\lceil even^{\star}\ 2 \rceil : \star \overset{\texttt{Bool}}{\Longrightarrow} \star \\
\longrightarrow \\
(odd\ 1) : \texttt{Bool} \overset{\texttt{Bool}}{\Longrightarrow} \star \\
\longrightarrow \\
\lceil even^{\star}\ 0 \rceil : \star \overset{\texttt{Bool}}{\Longrightarrow} \star \\
\longrightarrow \\
\lceil \texttt{true} \rceil
\end{array}
$$

Part XI

Coercions and Threesomes

Coercions and threesomes

Coercions	c, d	$::=$	$\mathtt{id}(A) \mid G! \mid p?G \mid c \to d \mid c\,;d$
Threesome coercions	s, t	$::=$	$\star \mid p?i \mid i$
Injection coercions	i	$::=$	$g! \mid g \mid \bot^{GpH}$
Ground coercions	g, h	$::=$	$\iota \mid s \to t$

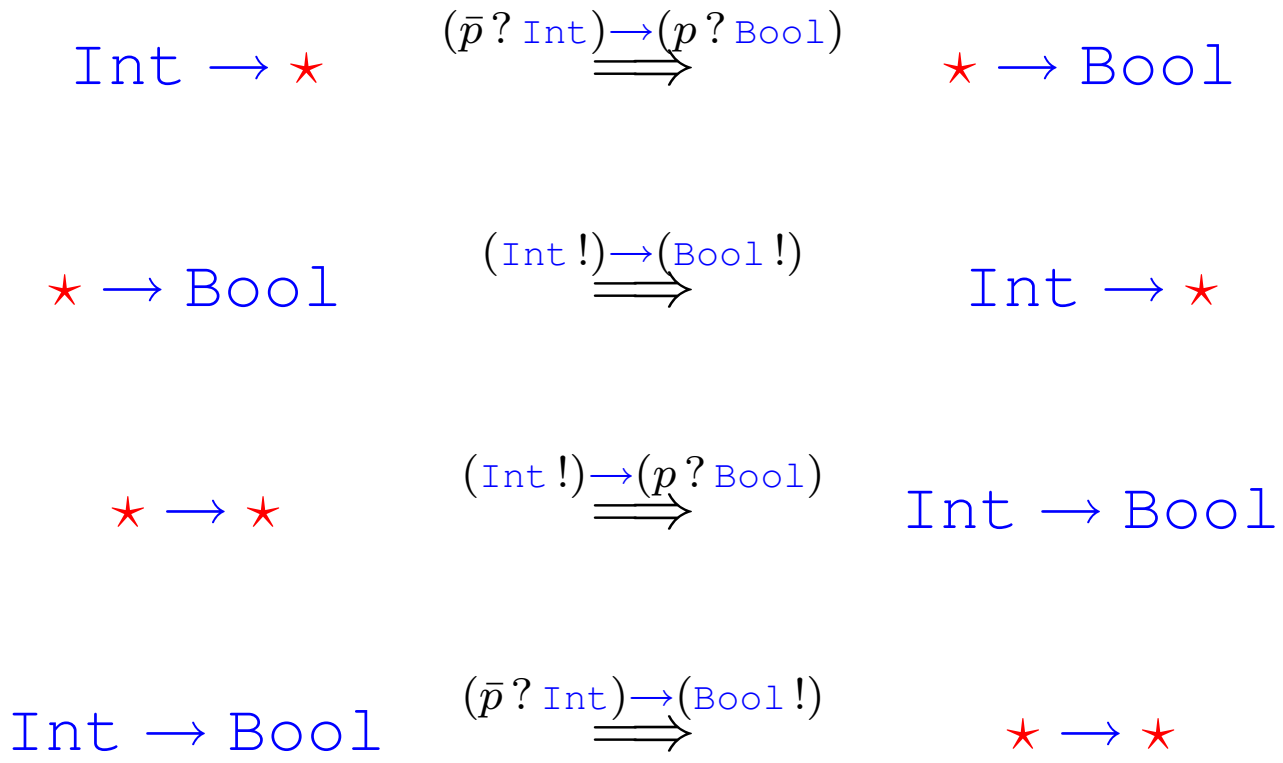
$$c : A \Rightarrow B$$

$$s : A \Rightarrow B$$

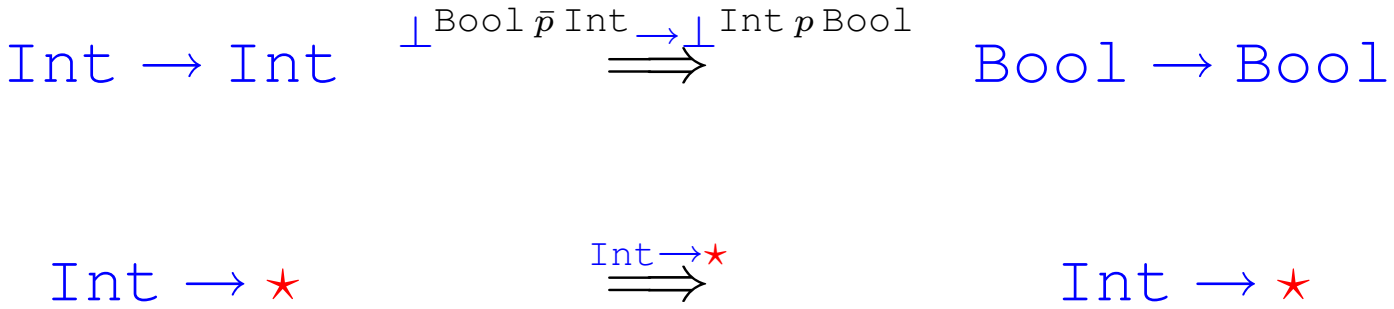
$$i : A \Rightarrow B \quad \text{implies } A \neq \star$$

$$g : A \Rightarrow B \quad \text{implies } A \neq \star \text{ and } B \neq \star$$

Threesomes are coercions



Threesomes are coercions



Threesome typing

$$\frac{}{\iota : \iota \Rightarrow \iota} \qquad \frac{}{\star : \star \Rightarrow \star}$$

$$\frac{g : A \Rightarrow G}{(g\,!) : A \Rightarrow \star}$$

$$\frac{i : G \Rightarrow A}{(p\,?\,i) : \star \Rightarrow A}$$

$$\frac{s : A' \Rightarrow A \quad t : B \Rightarrow B'}{(s \to t) : A \to B \Rightarrow A' \to B'}$$

$$\frac{\star \neq A \quad A \prec G \quad G \neq H}{(\bot^{GpH}) : A \Rightarrow B}$$

Threesome composition

$$
\begin{aligned}
\iota \fatsemi \iota &= \iota \\
(s \to t) \fatsemi (s' \to t') &= (s' \fatsemi s) \to (t \fatsemi t') \\
\star \fatsemi t &= t \\
(g\,!) \fatsemi \star &= g\,! \\
(p\,?\,i) \fatsemi t &= p\,?\,(i \fatsemi t) \\
g \fatsemi (h\,!) &= (g \fatsemi h)\,! \\
(g\,!) \fatsemi (p\,?\,i) &= \begin{cases} g \fatsemi i & \text{if } G = H \\ \bot^{GpH} & \text{if } G \neq H \end{cases} \quad \text{where} \quad \begin{array}{l} g : A \Rightarrow G \\ i : H \Rightarrow B \end{array} \\
\bot^{GpH} \fatsemi s &= \bot^{GpH} \\
g \fatsemi \bot^{GpH} &= \bot^{GpH}
\end{aligned}
$$

Part XII

Conclusion

Publications

- Wadler and Findler
  Well-typed programs can't be blamed
  ESOP 2009
- Siek and Wadler
  Threesomes, with and without blame
  POPL 2010
- Ahmed, Findler, Siek, and Wadler
  Blame for all
  POPL 2011
- Siek, Thiemann, Wadler
  Blame, coercions, and threesomes, precisely
  draft

A new slogan for type safety

Milner (1978):

Well-typed programs can't go wrong.

Felleisen and Wright (1994); Harper (2002):

Well-typed programs don't get stuck.

Wadler and Findler (2008):

Well-typed programs can't be blamed.