Well-typed programs can't be blamed

Philip Wadler

University of Edinburgh

A repeated theme

Findler and Felleisen (2002):

Contracts

Flanagan (2006):

Hybrid types

Siek and Taha (2006):

Gradual types

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

$$
\begin{array}{l}
\texttt{let} \\
\quad x = 2 \\
\quad f = \langle \texttt{Int} \to \texttt{Int} \Leftarrow \texttt{Dyn} \rangle^{pn} \lceil \lambda y.\, y + 1 \rceil \\
\quad h = \lambda g : \texttt{Int} \to \texttt{Int}.\, g\,(g\,x) \\
\texttt{in} \\
\quad h\,f \\
\longrightarrow \\
\quad 4 : \texttt{Int}
\end{array}
$$

A partly typed program—narrowing

$$
\begin{array}{l}
\texttt{let} \\
\quad x = 2 \\
\quad f = \langle \texttt{Int} \to \texttt{Int} \Leftarrow \texttt{Dyn} \rangle^{pn} \lceil \lambda y.\, \texttt{'b'} \rceil \\
\quad h = \lambda g : \texttt{Int} \to \texttt{Int}.\, g\,(g\,x) \\
\texttt{in} \\
\quad h\ f \\
\longrightarrow \\
\quad \texttt{blame}\, p
\end{array}
$$

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

Another partly typed program—widening

$$
\begin{array}{l}
\texttt{let} \\
\quad x = \lceil 2 \rceil \\
\quad f = \langle \texttt{Dyn} \Leftarrow \texttt{Int} \to \texttt{Int} \rangle^{pn}\ (\lambda y : \texttt{Int}.\ y + 1) \\
\quad h = \lceil \lambda g.\ g\ (g\ x) \rceil \\
\texttt{in} \\
\quad \lceil h\ f \rceil \\
\longrightarrow \\
\quad \lceil 4 \rceil
\end{array}
$$

Another partly typed program—widening

$$
\begin{array}{l}
\texttt{let} \\
\quad x = \lceil \texttt{'a'} \rceil \\
\quad f = \langle \texttt{Dyn} \Leftarrow \texttt{Int} \to \texttt{Int} \rangle^{pn}\ (\lambda y : \texttt{Int}.\ y + 1) \\
\quad h = \lceil \lambda g.\ g\ (g\ x) \rceil \\
\texttt{in} \\
\quad \lceil h\ f \rceil \\
\longrightarrow \\
\quad \texttt{blame}\ n
\end{array}
$$

Negative (contravariant): blame the context containg the cast

Untyped = Uni-typed

$$
\begin{array}{lcl}
\lceil x \rceil & = & x \\
\lceil n \rceil & = & \langle \mathtt{Dyn} \Leftarrow \mathtt{Int} \rangle\, n \\
\lceil \lambda x.\, N \rceil & = & \langle \mathtt{Dyn} \Leftarrow \mathtt{Dyn} \to \mathtt{Dyn} \rangle\, (\lambda x : \mathtt{Dyn}.\, \lceil N \rceil) \\
\lceil M\ N \rceil & = & (\langle \mathtt{Dyn} \to \mathtt{Dyn} \Leftarrow \mathtt{Dyn} \rangle\, \lceil M \rceil)\, \lceil N \rceil
\end{array}
$$

(slogan due to Bob Harper)

Blame

$$\langle \texttt{Int} \Leftarrow \texttt{Dyn} \rangle^{pn} \lceil \texttt{2} \rceil$$

$$\longrightarrow$$

$$\texttt{2}$$

$$\langle \texttt{Int} \Leftarrow \texttt{Dyn} \rangle^{pn} \lceil \texttt{'a'} \rceil$$

$$\longrightarrow$$

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

The Blame Game—widening

$$(\langle \texttt{Dyn} \to \texttt{Dyn} \Leftarrow \texttt{Int} \to \texttt{Int} \rangle^{pn}\ (\lambda y : \texttt{Int}.\, y + 1))\ \lceil 2 \rceil$$

$$\longrightarrow$$

$$\langle \texttt{Dyn} \Leftarrow \texttt{Int} \rangle^{pn}\ ((\lambda y : \texttt{Int}.\, y + 1)\ (\langle \texttt{Int} \Leftarrow \texttt{Dyn} \rangle^{np}\ \lceil 2 \rceil))$$

$$\longrightarrow$$

$$\lceil 3 \rceil$$

The Blame Game—widening

$$(\langle \texttt{Dyn} \to \texttt{Dyn} \Leftarrow \texttt{Int} \to \texttt{Int} \rangle^{pn}\ (\lambda y : \texttt{Int}.\, y + 1))\ \lceil \texttt{'a'} \rceil$$

$$\longrightarrow$$

$$\langle \texttt{Dyn} \Leftarrow \texttt{Int} \rangle^{pn}\ ((\lambda y : \texttt{Int}.\, y + 1)\ (\langle \texttt{Int} \Leftarrow \texttt{Dyn} \rangle^{np}\ \lceil \texttt{'a'} \rceil))$$

$$\longrightarrow$$

$$\texttt{blame}\ n$$

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

The Blame Game—narrowing

$$(\langle \texttt{Int} \to \texttt{Int} \Leftarrow \texttt{Dyn} \to \texttt{Dyn} \rangle^{pn}\ (\lambda y : \texttt{Dyn}.\ \lceil y + 1 \rceil))\ 2$$

$$\longrightarrow$$

$$\langle \texttt{Int} \Leftarrow \texttt{Dyn} \rangle^{pn}\ ((\lambda y : \texttt{Dyn}.\ \lceil y + 1 \rceil)\ (\langle \texttt{Dyn} \Leftarrow \texttt{Int} \rangle^{np}\ 2))$$

$$\longrightarrow$$

$$3$$

The Blame Game—narrowing

$$(\langle \texttt{Int} \to \texttt{Int} \Leftarrow \texttt{Dyn} \to \texttt{Dyn} \rangle^{pn}\ (\lambda y : \texttt{Dyn}.\ \lceil \texttt{'b'} \rceil))\ \texttt{2}$$

$$\longrightarrow$$

$$\langle \texttt{Int} \Leftarrow \texttt{Dyn} \rangle^{pn}\ ((\lambda y : \texttt{Dyn}.\ \lceil \texttt{'b'} \rceil)\ (\langle \texttt{Dyn} \Leftarrow \texttt{Int} \rangle^{np}\ \texttt{2}))$$

$$\longrightarrow$$

$$\texttt{blame}\ p$$

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

Contracts

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

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

Subtype

If $S <: T$ then $\langle T \Leftarrow S \rangle^{pn}\ s \not\longrightarrow \texttt{blame}\ p, \texttt{blame}\ n.$

Definition:

$$\frac{}{\texttt{Dyn} <: \texttt{Dyn}}$$

$$\frac{S' <: S \qquad T <: T'}{S \to T <: S' \to T'}$$

$$\frac{s \text{ implies } t}{\{x : B \mid s\} <: \{x : B \mid t\}}$$

Example:

$$\texttt{Nat} \to \texttt{Int} <: \texttt{Int} \to \texttt{Nat}$$

Positive subtype—widening

If $S <:^{+} T$ then $\langle T \Leftarrow S \rangle^{pn}\ s \not\longrightarrow \texttt{blame}\ p$.

Definition:

$$\frac{}{S <: \texttt{Dyn}}$$

$$\frac{S' <:^{-} S \qquad T <:^{+} T'}{S \to T <:^{+} S' \to T'}$$

$$\frac{s \text{ implies } t}{\{x : B \mid s\} <:^{+} \{x : B \mid t\}}$$

Examples:

$$\texttt{Int} \to \texttt{Int} <:^{+} \texttt{Dyn}$$

$$\texttt{Nat} \to \texttt{Nat} <:^{+} \texttt{Int} \to \texttt{Int}$$

Negative subtype—narrowing

If $S <:^{-} T$ then $\langle T \Leftarrow S \rangle^{pn}\ s \not\longrightarrow \texttt{blame}\ n$.

Definition:

$$\frac{}{\texttt{Dyn} <:^{-} T}$$

$$\frac{S' <:^{+} S \qquad T <:^{-} T'}{S \to T <:^{-} S' \to T'}$$

$$\frac{}{\{x : B \mid s\} <:^{-} \{x : B \mid t\}}$$

Examples:

$$\texttt{Dyn} <:^{-} \texttt{Int} \to \texttt{Int}$$

$$\texttt{Int} \to \texttt{Int} <:^{-} \texttt{Nat} \to \texttt{Nat}$$

A new slogan for type safety

Milner:

Well-typed programs can't go wrong.

Harper, Felleisen and Wright:

Well-typed programs don't get stuck.

Wadler and Findler:

Well-typed programs can't be blamed.