Type and Translation Rules for Arrow Notation in GHC

Ross Paterson   Simon Peyton Jones

September 16, 2004

GHC is full.
SIMON MARLOW (shortly before arrow notation was added)

Simple type rules

Let's assume that ordinary Haskell type judgements have the form $\Gamma \vdash e :: \tau$, ignoring type constraints for simplicity. The rules below define a new judgement of the form

$$\Gamma \mid \Delta \vdash_a c :: \tau_1 \rightharpoonup \cdots \; \tau_n \rightharpoonup \tau$$

The components of this judgement are as follows:

Γ an ordinary Haskell environment, for variables introduced outside the current **proc**-expression.

Δ typings of variables introduced in the current **proc**-expression.

a the arrow type of the command being constructed.

c a *command*: a new syntactic category. Many of the forms resemble expression forms, but have a different semantics.

τ_i the types of a *stack* of additional, anonymous inputs to the command. τ_1 is the type of the top element.

τ the type of values returned by the command.

Note that $\rightharpoonup$ is not a type constructor; τ and τ_i are ordinary Haskell types, but $\tau_1 \rightharpoonup \cdots \; \tau_n \rightharpoonup \tau$ is something else.

$$\frac{\begin{array}{l} p :: \tau \Leftrightarrow \Delta_p \\ \Gamma \mid \Delta_p \vdash_a c :: \tau' \end{array}}{\Gamma \vdash \mathbf{proc}\, p \rightarrow c :: a\ \tau\ \tau'}$$

$$\frac{\begin{array}{l} \Gamma \vdash f :: a\ (\tau, \bar{\tau})\ \tau' \\ \Gamma, \Delta \vdash e :: \tau \end{array}}{\Gamma \mid \Delta \vdash_a f \prec e :: \bar{\tau} \rightharpoonup \tau'}$$

$$\frac{\begin{array}{l} \Gamma, \Delta \vdash f :: a\ (\tau, \bar{\tau})\ \tau' \\ \Gamma, \Delta \vdash e :: \tau \end{array}}{\Gamma \mid \Delta \vdash_a f \prec\!\!\prec e :: \bar{\tau} \rightharpoonup \tau'}$$

$$\frac{\begin{array}{l} \Gamma, \Delta \vdash e :: \mathsf{Bool} \\ \Gamma \mid \Delta \vdash_a c_1 :: \bar{\tau} \rightharpoonup \tau \\ \Gamma \mid \Delta \vdash_a c_2 :: \bar{\tau} \rightharpoonup \tau \end{array}}{\Gamma \mid \Delta \vdash_a \mathbf{if}\ e\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2 :: \bar{\tau} \rightharpoonup \tau}$$

$$\frac{\Gamma \mid \Delta, \Delta_{binds} \vdash_a c :: \bar{\tau} \rightharpoonup \tau}{\Gamma \mid \Delta \vdash_a \mathbf{let}\ binds\ \mathbf{in}\ c :: \bar{\tau} \rightharpoonup \tau}$$

$$\frac{\begin{array}{l} p :: \tau \Leftrightarrow \Delta_p \\ \Gamma \mid \Delta, \Delta_p \vdash_a c :: \bar{\tau} \rightharpoonup \tau' \end{array}}{\Gamma \mid \Delta \vdash_a \lambda\, p \rightarrow c :: \tau \rightharpoonup \bar{\tau} \rightharpoonup \tau'}$$

$$\frac{\begin{array}{l} \Gamma \mid \Delta \vdash_a c :: \tau \rightharpoonup \bar{\tau} \rightharpoonup \tau' \\ \Gamma, \Delta \vdash e :: \tau \end{array}}{\Gamma \mid \Delta \vdash_a c\ e :: \bar{\tau} \rightharpoonup \tau'}$$

$$\frac{\begin{array}{l} \Gamma \vdash e :: \forall w.\, a_i\ (w, \bar{\tau}_i)\ \tau_i \rightarrow a\ (w, \bar{\tau})\ \tau \\ \Gamma \mid \Delta \vdash_{a_i} c_i :: \bar{\tau}_i \rightharpoonup \tau_i \end{array}}{\Gamma \mid \Delta \vdash_a (\!|e\ c_1\ \ldots\ c_n|\!) :: \bar{\tau} \rightharpoonup \tau}$$

$$\frac{\Gamma \mid \Delta \vdash_a c :: \tau}{\Gamma \mid \Delta \vdash_a \mathbf{do}\ \{\ c\ \} :: \tau}$$

$$\frac{\Gamma \mid \Delta, \Delta_{binds} \vdash_a \mathbf{do}\ \{\ s\ \} :: \tau}{\Gamma \mid \Delta \vdash_a \mathbf{do}\ \{\ \mathbf{let}\ binds; s\ \} :: \tau}$$

$$\frac{\begin{array}{l} \Gamma \mid \Delta \vdash_a c :: \tau \\ \Gamma \mid \Delta \vdash_a \mathbf{do}\ \{\ s\ \} :: \tau' \end{array}}{\Gamma \mid \Delta \vdash_a \mathbf{do}\ \{\ c; s\ \} :: \tau'}$$

$$\frac{\begin{array}{l} p :: \tau \Leftrightarrow \Delta_p \\ \Gamma \mid \Delta \vdash_a c :: \tau \\ \Gamma \mid \Delta, \Delta_p \vdash_a \mathbf{do}\ \{\ s\ \} :: \tau' \end{array}}{\Gamma \mid \Delta \vdash_a \mathbf{do}\ \{\ p \leftarrow c; s\ \} :: \tau'}$$

Typing and translation rules for arrow notation

These rules take a judgement

$$\Gamma \mid \Delta \vdash_a c :: \tau_1 \rightharpoonup \cdots\ \tau_n \rightharpoonup \tau$$

and produce an expression c^* such that

$$\Gamma \vdash c^* :: a\ (\ldots((\tau_\Delta), \tau_1), \ldots, \tau_n)\ \tau$$

where (τ_Δ) means a tuple of the types of Δ.

$$
\begin{array}{lcl}
p :: \tau \Rightarrow \Delta & \longmapsto & k = \lambda\, p \rightarrow (\Delta) \\
\Gamma \mid \Delta \vdash_a c :: \tau' & \longmapsto & c^* \\
\hline
\Gamma \vdash \mathbf{proc}\; p \rightarrow c :: a\; \tau\; \tau' & \longmapsto & \mathsf{arr}\; k \ggg c^*
\end{array}
$$

$$
\begin{array}{lcl}
\Gamma \vdash f :: a\; (\tau, \bar{\tau})\; \tau' & \longmapsto & f \\
\Gamma, \Delta \vdash e :: \tau & & \\
\hline
\Gamma \mid \Delta \vdash_a f \mathrel{-\!\!\prec} e :: \bar{\tau} \rightharpoonup \tau' & \longmapsto & \mathsf{arr}\; (\lambda\, ((\Delta), \bar{\tau}) \rightarrow (e, \bar{\tau})) \ggg f
\end{array}
$$

$$
\begin{array}{lcl}
\Gamma \vdash f :: a\; (\tau, \bar{\tau})\; \tau' & & \\
\Gamma, \Delta \vdash e :: \tau & & \\
\hline
\Gamma \mid \Delta \vdash_a f \mathrel{-\!\!\prec\!\!\prec} e :: \bar{\tau} \rightharpoonup \tau' & \longmapsto & \mathsf{arr}\; (\lambda\, ((\Delta), \bar{\tau}) \rightarrow (f, (e, \bar{\tau}))) \ggg \mathsf{app}
\end{array}
$$

$$
\begin{array}{lcl}
\Gamma, \Delta \vdash e :: \mathsf{Bool} & \longmapsto & k = \lambda\, ((\Delta), \bar{\tau}) \rightarrow \mathbf{if}\; e\; \mathbf{then}\; \mathsf{Left}\; ((\Delta_1), \bar{\tau}) \\
 & & \qquad\qquad\qquad\qquad \mathbf{else}\; \mathsf{Right}\; ((\Delta_2), \bar{\tau}) \\
\Gamma \mid \Delta_1 \vdash_a c_1 :: \bar{\tau} \rightharpoonup \tau & \longmapsto & c_1^* \\
\Gamma \mid \Delta_2 \vdash_a c_2 :: \bar{\tau} \rightharpoonup \tau & \longmapsto & c_2^* \\
\hline
\Gamma \mid \Delta \vdash_a \mathbf{if}\; e\; \mathbf{then}\; c_1\; \mathbf{else}\; c_2 :: \bar{\tau} \rightharpoonup \tau & \longmapsto & \mathsf{arr}\; k \ggg (c_1^* \mathbin{|||} c_2^*)
\end{array}
$$

$$
\begin{array}{lcl}
\Delta, \mathit{binds} \Rightarrow \Delta' & \longmapsto & k = \lambda\, ((\Delta), \bar{\tau}) \rightarrow \mathbf{let}\; \mathit{binds}\; \mathbf{in}\; ((\Delta'), \bar{\tau}) \\
\Gamma \mid \Delta' \vdash_a c :: \bar{\tau} \rightharpoonup \tau & \longmapsto & c^* \\
\hline
\Gamma \mid \Delta \vdash_a \mathbf{let}\; \mathit{binds}\; \mathbf{in}\; c :: \bar{\tau} \rightharpoonup \tau & \longmapsto & \mathsf{arr}\; k \ggg c^*
\end{array}
$$

$$
\begin{array}{lcl}
\Delta, p :: \tau \Rightarrow \Delta' & \longmapsto & k = \lambda\, (((\Delta), p), \bar{\tau}) \rightarrow ((\Delta'), \bar{\tau}) \\
\Gamma \mid \Delta' \vdash_a c :: \bar{\tau} \rightharpoonup \tau' & \longmapsto & c^* \\
\hline
\Gamma \mid \Delta \vdash_a \lambda\, p \rightarrow c :: \tau \rightharpoonup \bar{\tau} \rightharpoonup \tau' & \longmapsto & \mathsf{arr}\; k \ggg c^*
\end{array}
$$

$$
\begin{array}{lcl}
\Gamma \mid \Delta' \vdash_a c :: \tau \rightharpoonup \bar{\tau} \rightharpoonup \tau' & \longmapsto & c^* \\
\left.\begin{array}{l} \Delta' \subseteq \Delta \\ \Gamma, \Delta \vdash e :: \tau \end{array}\right\} & \longmapsto & k = \lambda\, ((\Delta), \bar{\tau}) \rightarrow (((\Delta'), e), \bar{\tau}) \\
\hline
\Gamma \mid \Delta \vdash_a c\; e :: \bar{\tau} \rightharpoonup \tau' & \longmapsto & \mathsf{arr}\; k \ggg c^*
\end{array}
$$

$$
\begin{array}{lcl}
\Delta_i \subseteq \Delta & \longmapsto & k_i = \lambda\, ((\Delta), \bar{\tau}_i) \rightarrow ((\Delta_i), \bar{\tau}_i) \\
\Gamma \vdash e :: \forall\, w.\; a_i\; (w, \bar{\tau}_i)\; \tau_i \rightarrow a\; (w, \bar{\tau})\; \tau & \longmapsto & e \\
\Gamma \mid \Delta_i \vdash_{a_i} c_i :: \bar{\tau}_i \rightharpoonup \tau_i & \longmapsto & c_i^* \\
\hline
\Gamma \mid \Delta \vdash_a (\!|\, e\; c_1\; \ldots\; c_n \,|\!) :: \bar{\tau} \rightharpoonup \tau & \longmapsto & e_{(\Delta)}\; (\mathsf{arr}\; k_i \ggg c_i^*)
\end{array}
$$

Do notation

$$
\begin{array}{l c l}
\Gamma \mid \Delta \vdash_a c :: \tau & \longmapsto & c^* \\ \hline
\Gamma \mid \Delta \vdash_a \mathbf{do}\ \{\ c\ \} :: \tau & \longmapsto & c^*
\end{array}
$$

$$
\begin{array}{l c l}
\Delta, \mathit{binds} \Rightarrow \Delta' & \longmapsto & k = \lambda\,(\Delta) \rightarrow \mathbf{let}\ \mathit{binds}\ \mathbf{in}\ (\Delta') \\
\Gamma \mid \Delta' \vdash_a \mathbf{do}\ \{\ s\ \} :: \tau & \longmapsto & s^* \\ \hline
\Gamma \mid \Delta \vdash_a \mathbf{do}\ \{\ \mathbf{let}\ \mathit{binds}; s\ \} :: \tau & \longmapsto & \mathsf{arr}\ k \ggg s^*
\end{array}
$$

$$
\begin{array}{l c l}
\Delta \subseteq \Delta_1 \cup \Delta_2 & \longmapsto & k = \lambda\,(\Delta) \rightarrow ((\Delta_1), (\Delta_2)) \\
\Gamma \mid \Delta_1 \vdash_a c :: \tau & \longmapsto & c^* \\
\Gamma \mid \Delta_2 \vdash_a \mathbf{do}\ \{\ s\ \} :: \tau' & \longmapsto & s^* \\ \hline
\Gamma \mid \Delta \vdash_a \mathbf{do}\ \{\ c; s\ \} :: \tau' & \longmapsto & \mathsf{arr}\ k \ggg \mathsf{first}\ c^* \ggg \mathsf{arr}\ \mathsf{snd} \ggg s^*
\end{array}
$$

$$
\begin{array}{l c l}
\Delta \subseteq \Delta_1 \cup \Delta_2 & \longmapsto & k = \lambda\,(\Delta) \rightarrow ((\Delta_1), (\Delta_2)) \\
\Gamma \mid \Delta_1 \vdash_a c :: \tau & \longmapsto & c^* \\
p :: \tau, \Delta_2 \Rightarrow \Delta' & \longmapsto & k' = \lambda\,(p, (\Delta_2)) \rightarrow (\Delta') \\
\Gamma \mid \Delta' \vdash_a \mathbf{do}\ \{\ s\ \} :: \tau' & \longmapsto & s^* \\ \hline
\Gamma \mid \Delta \vdash_a \mathbf{do}\ \{\ p \leftarrow c; s\ \} :: \tau' & \longmapsto & \mathsf{arr}\ k \ggg \mathsf{first}\ c^* \ggg \mathsf{arr}\ k' \ggg s^*
\end{array}
$$