Erratum

A Partial evaluator for the untyped lambda-calculus by Carsten K. Gomard and Neil D. Jones. Due to a typesetting error, figure 2, page 31 of volume 1 part 1 was incorrect. The correct version appears below. The editors apologise for the error.

Semantic domains

$$
\begin{aligned}
&\mathit{2Val} = \mathit{Base} + \mathit{2Funval} + \mathit{Code}\\
&\mathit{2Funval} = \mathit{2Val} \to \mathit{2Val}\\
&\mathit{Code} = \mathit{Expression}\ (= \text{the set of one-level expressions})\\
&\mathit{2Env} = \mathit{Var} \to \mathit{2Val}
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{T} : \mathit{2Expression} \to \mathit{2Env} \to \mathit{2Val}\\
&\mathcal{T}[\![\mathsf{var}]\!]\rho = \rho(\mathsf{var})
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{T}[\![\lambda\mathsf{var}.\mathsf{texp}]\!]\rho = (\lambda \mathit{value}.(\mathcal{T}[\![\mathsf{texp}]\!]\rho[\mathsf{var} \mapsto \mathit{value}])) \uparrow \mathit{2Funval}\\
&\mathcal{T}[\![\mathsf{texp}_1\ @\ \mathsf{texp}_2]\!]\rho = (\mathcal{T}[\![\mathsf{texp}_1]\!]\rho \downarrow \mathit{2Funval})\ (\mathcal{T}[\![\mathsf{texp}_2]\!]\rho)\\
&\mathcal{T}[\![\mathsf{fix}\ \mathsf{texp}]\!]\rho = \mathit{fix}\ (\mathcal{T}[\![\mathsf{texp}]\!]\rho \downarrow \mathit{2Funval})\\
&\mathcal{T}[\![\mathsf{if}\ \mathsf{texp}_1\ \mathsf{texp}_2\ \mathsf{texp}_3]\!]\rho = (\mathcal{T}[\![\mathsf{texp}_1]\!]\rho \downarrow \mathit{Base}) \to \mathcal{T}[\![\mathsf{texp}_2]\!]\rho, \mathcal{T}[\![\mathsf{texp}_3]\!]\rho\\
&\mathcal{T}[\![\mathsf{const}\ \mathsf{c}]\!]\rho = c \uparrow \mathit{Base}\\
&\mathcal{T}[\![\mathsf{lift}\ \mathsf{texp}]\!]\rho = \mathit{build\text{-}const}\,(\mathcal{T}[\![\mathsf{texp}]\!]\rho \downarrow \mathit{Base}) \uparrow \mathit{Code}
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{T}[\![\underline{\lambda}\mathsf{var}.\mathsf{texp}]\!]\rho = \text{let } \mathit{nvar} = \mathit{newname} \text{ in}\\
&\qquad\qquad \mathit{build\text{-}\lambda}(\mathit{nvar}, \mathcal{T}[\![\mathsf{texp}]\!]\rho[\mathsf{var} \mapsto \mathit{nvar}] \downarrow \mathit{Code}) \uparrow \mathit{Code}\\
&\mathcal{T}[\![\mathsf{texp}_1\ \underline{@}\ \mathsf{texp}_2]\!]\rho = \mathit{build\text{-}@}(\mathcal{T}[\![\mathsf{texp}_1]\!]\rho \downarrow \mathit{Code}, \mathcal{T}[\![\mathsf{texp}_2]\!]\rho \downarrow \mathit{Code}) \uparrow \mathit{Code}\\
&\mathcal{T}[\![\underline{\mathsf{fix}}\ \mathsf{texp}]\!]\rho = \mathit{build\text{-}fix}\,(\mathcal{T}[\![\mathsf{texp}]\!]\rho \downarrow \mathit{Code}) \uparrow \mathit{Code}\\
&\mathcal{T}[\![\underline{\mathsf{if}}\ \mathsf{texp}_1\ \mathsf{texp}_2\ \mathsf{texp}_3]\!]\rho = \mathit{build\text{-}if}\ (\mathcal{T}[\![\mathsf{texp}_1]\!]\rho \downarrow \mathit{Code},\\
&\qquad\qquad\qquad\qquad \mathcal{T}[\![\mathsf{texp}_2]\!]\rho \downarrow \mathit{Code},\\
&\qquad\qquad\qquad\qquad \mathcal{T}[\![\mathsf{texp}_3]\!]\rho \downarrow \mathit{Code}) \uparrow \mathit{Code}\\
&\mathcal{T}[\![\underline{\mathsf{const}}\ \mathsf{c}]\!]\rho = \mathit{build\text{-}const}\,(c) \uparrow \mathit{Code}
\end{aligned}
$$

Fig. 2. Two-level lambda calculus semantics.

https://doi.org/10.1017/S0956796800000150 Published online by Cambridge University Press