GADTs Meet Their Match:

Pattern-Matching Warnings That Account for GADTs, Guards, and Laziness

SEBASTIAN GRAF, Karlsruhe Institute of Technology, Germany
SIMON PEYTON JONES, Microsoft Research, UK

Authors' addresses: Sebastian Graf, Karlsruhe Institute of Technology, Karlsruhe, Germany, sebastian.graf@kit.edu; Simon Peyton Jones, Microsoft Research, Cambridge, UK, simonpj@microsoft.com.

Guard Syntax

$$
\begin{array}{rlll}
K \in & \text{Con} & & \\
x, y, a, b \in & \text{Var} & & \\
\tau, \sigma \in & \text{Type} & & \\
e \in & \text{Expr} & ::= & x : \tau \\
& & \mid & K\ \overline{\tau}\ \overline{\gamma}\ \overline{e : \tau} \\
& & \mid & \ldots
\end{array}
\qquad
\begin{array}{rlll}
n \in & \mathbb{N} & & \\
\gamma \in & \text{TyCt} & ::= & \tau_1 \sim \tau_2 \mid \ldots \\
p \in & \text{Pat} & ::= & \_ \\
& & \mid & K\ \overline{y} \\
& & \mid & \ldots \\
g \in & \text{Grd} & ::= & \text{let } x : \tau = e \\
& & \mid & K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow x \\
& & \mid & !x
\end{array}
$$

Constraint Formula Syntax

$$
\begin{array}{lcll}
\Gamma & ::= & \varnothing \mid \Gamma, x : \tau \mid \Gamma, a & \text{Context} \\
\delta & ::= & \checkmark \mid \times \mid K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow x \mid x \not\approx K \mid x \approx \bot \mid x \not\approx \bot \mid x \approx e & \text{Constraint Literals} \\
\Delta & ::= & \delta \mid \Delta \wedge \Delta \mid \Delta \vee \Delta & \text{Formula} \\
\nabla & ::= & \varnothing \mid \nabla, \delta & \text{Inert Set}
\end{array}
$$

Clause Tree Syntax

$$
\begin{array}{rcl}
t_G, u_G \in \text{Gdt} & ::= & \texttt{Rhs}\ n \mid t_G; u_G \mid \texttt{Guard}\ g\ t_G \\
t_A, u_A \in \text{Ant} & ::= & \texttt{AccessibleRhs}\ n \mid \texttt{InaccessibleRhs}\ n \mid t_A; u_A \mid \texttt{MayDiverge}\ t_A
\end{array}
$$

Checking Guard Trees

$$\boxed{\mathcal{U}(\Delta, t_G) = \Delta}$$

$$
\begin{array}{lcl}
\mathcal{U}(\Delta, \texttt{Rhs}\ n) & = & \times \\
\mathcal{U}(\Delta, (t; u)) & = & \mathcal{U}(\mathcal{U}(\Delta, t), u) \\
\mathcal{U}(\Delta, \texttt{Guard}\ (!x)\ t) & = & \mathcal{U}(\Delta \wedge (x \not\approx \bot), t) \\
\mathcal{U}(\Delta, \texttt{Guard}\ (\text{let } x = e)\ t) & = & \mathcal{U}(\Delta \wedge (x \approx e), t) \\
\mathcal{U}(\Delta, \texttt{Guard}\ (K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow x)\ t) & = & (\Delta \wedge (x \not\approx K) \wedge (x \not\approx \bot)) \vee \mathcal{U}(\Delta \wedge (K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow x), gs)
\end{array}
$$

$$\boxed{\mathcal{A}_\Gamma(\Delta, t_G) = t_A}$$

$$
\begin{array}{lcl}
\mathcal{A}_\Gamma(\Delta, \texttt{Rhs}\ n) & = & \begin{cases} \texttt{InaccessibleRhs}\ n, & \mathcal{M}(\Gamma, \Delta) = \emptyset \\ \texttt{AccessibleRhs}\ n, & \text{otherwise} \end{cases} \\
\mathcal{A}_\Gamma(\Delta, (t; u)) & = & \mathcal{A}_\Gamma(\Delta, t); \mathcal{A}_\Gamma(\mathcal{U}(\Delta, t), u) \\
\mathcal{A}_\Gamma(\Delta, \texttt{Guard}\ (!x)\ t) & = & \begin{cases} \mathcal{A}_\Gamma(\Delta \wedge (x \not\approx \bot), t), & \mathcal{M}(\Gamma, \Delta \wedge (x \approx \bot)) = \emptyset \\ \texttt{MayDiverge}\ \mathcal{A}_\Gamma(\Delta \wedge (x \not\approx \bot), t) & \text{otherwise} \end{cases} \\
\mathcal{A}_\Gamma(\Delta, \texttt{Guard}\ (\text{let } x = e)\ t) & = & \mathcal{A}_\Gamma(\Delta \wedge (x \approx e), t) \\
\mathcal{A}_\Gamma(\Delta, \texttt{Guard}\ (K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow x)\ t) & = & \mathcal{A}_\Gamma(\Delta \wedge (K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow x), t)
\end{array}
$$

Putting it all together

(0) Input: Context with match vars Γ and desugared Gdt t

(1) Report n value vectors of $\mathcal{M}(\Gamma, \mathcal{U}(\checkmark, t))$ as uncovered

(2) Report the collected redundant and not-redundant-but-inaccessible clauses in $\mathcal{A}_\Gamma(\checkmark, t)$ (TODO: Write a function that collects the RHSs).

Add a constraint to the inert set

$$\boxed{\Gamma \vartriangleright \nabla \oplus \delta = \Gamma \vartriangleright \nabla}$$

$$
\begin{array}{lcl}
\Gamma \vartriangleright \nabla \oplus \times & = & \bot \\
\Gamma \vartriangleright \nabla \oplus \checkmark & = & \Gamma \vartriangleright \nabla \\
\Gamma \vartriangleright \nabla \oplus \gamma & = & \begin{cases} \Gamma \vartriangleright (\nabla, \gamma) & \text{if type checker deems } \gamma \text{ compatible with } \nabla \\ & \text{and } \forall x \in \mathrm{fvs}(\Gamma) : \Gamma \vartriangleright (\nabla, \gamma) \vdash x \\ \bot & \text{otherwise} \end{cases} \\
\Gamma \vartriangleright \nabla \oplus K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow x & = & \begin{cases} \Gamma, \overline{a}, \overline{y : \tau} \vartriangleright \nabla \oplus \overline{a \sim b} \oplus \overline{\gamma} \oplus \overline{y \approx z} & \text{if } K\ \overline{b}\ \overline{\gamma}\ \overline{z : \tau} \leftarrow x \in \nabla \\ \Gamma' \vartriangleright (\nabla', K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow x) & \text{where } \Gamma' \vartriangleright \nabla' = \Gamma, \overline{a}, \overline{y : \tau} \vartriangleright \nabla \oplus \overline{\gamma} \\ & \text{and } x \not\approx K \notin \nabla \\ & \text{and } \overline{\Gamma' \vartriangleright \nabla' \vdash y} \\ \bot & \text{otherwise} \end{cases} \\
\Gamma \vartriangleright \nabla \oplus x \not\approx K & = & \begin{cases} \bot & \text{if } K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow x \in \nabla \\ \bot & \text{if } x : \tau \in \Gamma \\ & \text{and } \forall K' : \sigma \in \mathrm{Cons}(\Gamma \vartriangleright \nabla, \tau) : x \not\approx K' \in (\nabla, x \not\approx K) \\ \bot & \text{if not } \Gamma \vartriangleright (\nabla, x \not\approx K) \vdash x \\ \Gamma \vartriangleright (\nabla, x \not\approx K) & \text{otherwise} \end{cases} \\
\Gamma \vartriangleright \nabla \oplus x \approx \bot & = & \begin{cases} \bot & \text{if } x \not\approx \bot \in \nabla \\ \Gamma \vartriangleright (\nabla, x \approx \bot) & \text{otherwise} \end{cases} \\
\Gamma \vartriangleright \nabla \oplus x \not\approx \bot & = & \begin{cases} \bot & \text{if } x \approx \bot \in \nabla \\ \bot & \text{if not } \Gamma \vartriangleright (\nabla, x \not\approx \bot) \vdash x \\ \Gamma \vartriangleright (\nabla, x \not\approx \bot) & \text{otherwise} \end{cases} \\
\Gamma \vartriangleright \nabla \oplus x \approx y & = & \begin{cases} \Gamma \vartriangleright \nabla & \text{if } \nabla(x) = z = \nabla(y) \\ \Gamma \vartriangleright \nabla, x \approx y \oplus \bigwedge \{ \delta \in \nabla \cap x \mid \text{x in } \delta \text{ renamed to y} \} & \text{if } \nabla(x) \neq z \text{ or } \nabla(y) \neq z \end{cases} \\
\Gamma \vartriangleright \nabla \oplus x \approx K\ \overline{\tau}\ \overline{\gamma}\ \overline{e'} & = & \Gamma, \overline{a}, \overline{y : \mathit{todo}} \vartriangleright \nabla \oplus K\ \overline{a}\ \overline{\gamma}\ \overline{y} \leftarrow x \oplus \overline{a \sim \tau} \oplus \overline{y \approx e'} \text{ where } \overline{a \# \Gamma}, \overline{y : \mathit{todo} \# (\Gamma, \overline{a})} \\
\Gamma \vartriangleright \nabla \oplus x \approx e & = & \Gamma \vartriangleright \nabla
\end{array}
$$

Test if x is inhabited considering ∇

$$\boxed{\Gamma \vartriangleright \nabla \vdash x}$$

$$
\frac{(\Gamma \vartriangleright \nabla \oplus x \approx \bot) \neq \bot}{\Gamma \vartriangleright \nabla \vdash x}
\qquad
\frac{\begin{array}{c} x : \tau \in \Gamma \quad K : \sigma \in \mathrm{Cons}(\Gamma \vartriangleright \nabla, \tau) \\ \mathit{instantiate} \\ (\Gamma, \overline{y : \tau'} \vartriangleright \nabla \oplus K\ \overline{a}\ \overline{\gamma}\ \overline{y} \leftarrow x) \neq \bot \end{array}}{\Gamma \vartriangleright \nabla \vdash x}
$$

Construct inhabited ∇s from Δ

$$\boxed{\mathcal{M}(\Gamma, \Delta) = \mathcal{P}(\overline{p})}$$

$$\boxed{\mathcal{M}(\Gamma \triangleright \nabla, \Delta) = \mathcal{P}(\Gamma \triangleright \nabla)}$$

$$\begin{array}{lcl}
\mathcal{M}(\Gamma, \Delta) & = & \bigcup \{ \mathcal{I}(\Gamma' \triangleright \nabla', \mathrm{fvs}(\Gamma)) \mid \forall (\Gamma' \triangleright \nabla') \in \mathcal{M}(\Gamma \triangleright \varnothing, \Delta) \} \\
\mathcal{M}(\Gamma \triangleright \nabla, \delta) & = & \begin{cases} \{\Gamma' \triangleright \nabla'\} & \text{where } \Gamma' \triangleright \nabla' = \Gamma \triangleright \nabla \oplus \delta \\ \emptyset & \text{otherwise} \end{cases} \\
\mathcal{M}(\Gamma \triangleright \nabla, \Delta_1 \wedge \Delta_2) & = & \bigcup \{ \mathcal{M}(\Gamma' \triangleright \nabla', \Delta_2) \mid \forall (\Gamma' \triangleright \nabla') \in \mathcal{M}(\Gamma \triangleright \nabla, \Delta_1) \} \\
\mathcal{M}(\Gamma \triangleright \nabla, \Delta_1 \vee \Delta_2) & = & \mathcal{M}(\Gamma \triangleright \nabla, \Delta_1) \cup \mathcal{M}(\Gamma \triangleright \nabla, \Delta_2)
\end{array}$$

Expand variables to Pat with ∇

$$\boxed{\mathcal{I}(\Gamma \triangleright \nabla, \overline{x}) = \mathcal{P}(\overline{p})}$$

$$\begin{array}{lcl}
\mathcal{I}(\Gamma \triangleright \nabla, \epsilon) & = & \{\epsilon\} \\
\mathcal{I}(\Gamma \triangleright \nabla, x_1 \ldots x_n) & = & \begin{cases} \{ (K\ q_1 \ldots q_m)\, p_2 \ldots p_n \mid \forall (q_1 \ldots q_m\, p_2 \ldots p_n) \in \mathcal{I}(\Gamma \triangleright \nabla, y_1 \ldots y_m x_2 \ldots x_n) \} & \text{if } K\ \overline{a}\ \overline{\gamma}\ \overline{y : \tau} \leftarrow \\ \{ _\ p_2 \ldots p_n \mid \forall (p_2 \ldots p_n) \in \mathcal{I}(\Gamma \triangleright \nabla, x_2 \ldots x_n) \} & \text{otherwise} \end{cases}
\end{array}$$