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

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

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

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

Checking Guard Trees

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

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

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

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

Putting it all together

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

(1) Report n pattern vectors of $\mathcal{G}(\Gamma, \mathcal{U}(\checkmark, t))$ 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).

Generate inhabitants of Δ

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

$$\mathcal{G}(\Gamma, \Delta) = \bigcup \{\mathcal{E}(\Gamma' \triangleright \nabla', \mathrm{fvs}(\Gamma)) \mid \forall (\Gamma' \triangleright \nabla') \in \mathcal{C}(\Gamma \triangleright \varnothing, \Delta)\}$$

Construct inhabited ∇s from Δ

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

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

Expand variables to Pat with ∇

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

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

Add a constraint to the inert set

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

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

$$\boxed{\nabla \cap x = \nabla}$$

$$
\begin{array}{lcl}
\varnothing \cap x & = & \varnothing \\
(\nabla, K\ \overline{a}\ \overline{\gamma}\ \overline{y} \leftarrow x) \cap x & = & (\nabla \cap x), K\ \overline{a}\ \overline{\gamma}\ \overline{y} \leftarrow x \\
(\nabla, x \not\approx K) \cap x & = & (\nabla \cap x), x \not\approx K \\
(\nabla, x \approx \bot) \cap x & = & (\nabla \cap x), x \approx \bot \\
(\nabla, x \not\approx \bot) \cap x & = & (\nabla \cap x), x \not\approx \bot \\
(\nabla, x \approx e) \cap x & = & (\nabla \cap x), x \approx e \\
(\nabla, \delta) \cap x & = & \nabla \cap x
\end{array}
$$

Test if x is inhabited considering ∇

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

$$\frac{(\Gamma \triangleright \nabla \oplus x \approx \bot) \neq \bot}{\Gamma \triangleright \nabla \vdash x} \qquad \frac{x : \tau \in \Gamma \quad K \in \mathsf{Cons}(\Gamma \triangleright \nabla, \tau) \quad \mathsf{Inst}(\Gamma, x, K) = \overline{\delta} \quad (\Gamma, \overline{y : \tau'} \triangleright \nabla \oplus \overline{\delta}) \neq \bot}{\Gamma \triangleright \nabla \vdash x}$$

$$\frac{x : \tau \in \Gamma \quad \mathsf{Cons}(\Gamma \triangleright \nabla, \tau) = \bot}{\Gamma \triangleright \nabla \vdash x} \qquad \frac{x : \tau \in \Gamma \quad K \in \mathsf{Cons}(\Gamma \triangleright \nabla, \tau) \quad \mathsf{Inst}(\Gamma, x, K) = \bot}{\Gamma \triangleright \nabla \vdash x}$$

Find data constructors of τ

$$\boxed{\mathsf{Cons}(\Gamma \triangleright \nabla, \tau) = \overline{K}}$$

$$\mathsf{Cons}(\Gamma \triangleright \nabla, \tau) = \begin{cases} \overline{K} & \tau = T\ \overline{\sigma} \text{ and } T \text{ data type with constructors } \overline{K} \\ & \text{(after normalisation according to the type constraints in } \nabla) \\ \bot & \text{otherwise} \end{cases}$$

Instantiate x to data constructor K

$$\boxed{\mathsf{Inst}(\Gamma, x, K) = \overline{\gamma}}$$

$$\mathsf{Inst}(\Gamma, x, K) = \begin{cases} \tau_x \sim \tau, K\ \overline{a}\ \overline{\gamma}\ \overline{y} \leftarrow x, \overline{y' \not\approx \bot} & K : \forall \overline{a}.\overline{\gamma} \Rightarrow \overline{\sigma} \rightarrow \tau, \overline{y} \# \Gamma, \overline{a} \# \Gamma, x : \tau_x \in \Gamma, \overline{y'} \text{ bind strict fields} \\ \bot & \text{otherwise} \end{cases}$$