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

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

Constraint Formula Syntax

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

Clause Tree Syntax

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

Checking Guard Trees

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

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

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

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

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).

Construct inhabited ∇s from Δ

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

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

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

Construct inhabited ∇s from Δ

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

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

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

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 \text{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 } \underline{x \not\approx K \notin \nabla} \\ & \text{and } \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' : \sigma \in \text{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 \bigwedge_{3} \{\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 \triangleright \nabla \oplus x \approx K\ \overline{\tau}\ \overline{\gamma}\ \overline{e'} & = & \Gamma, \overline{a}, \overline{y : todo} \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 : todo \# (\Gamma, \overline{a})} \\
\Gamma \triangleright \nabla \oplus x \approx e & = & \Gamma \triangleright \nabla
\end{array}$$

Test if x is inhabited considering ∇

This figure is completely out of date, don't waste your time

Test if Oracle state Delta is unsatisfiable

$$\boxed{\nvdash_{\textsc{Sat}} \Gamma \vdash \Delta}$$

$$\frac{\nvdash_{\textsc{Sat}} \Gamma \vdash fvs\Gamma \triangleright \Delta}{\nvdash_{\textsc{Sat}} \Gamma \vdash \Delta}$$

Test a list of SAT roots for inhabitants

$$\boxed{\nvdash_{\textsc{Sat}} \Gamma \vdash \overline{x} \triangleright \Delta}$$

$$\frac{\nvdash_{\textsc{Sat}} \Gamma \vdash x_i \triangleright \Delta}{\nvdash_{\textsc{Sat}} \Gamma \vdash \overline{x} \triangleright \Delta}$$

Test a single SAT root for inhabitants

$$\boxed{\nvdash_{\textsc{Sat}} \Gamma \vdash x \triangleright \Delta}$$

$$\frac{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta x \approx \bot \quad \{\overline{K}\} \text{ COMPLETE set} \quad \overline{\forall \overline{y : \tau}.\ \nvdash_{\textsc{Sat}} \Gamma, \overline{y : \tau} \vdash \wedge\Delta x \approx K\ \overline{y}}}{\nvdash_{\textsc{Sat}} \Gamma \vdash x \triangleright \Delta}$$

Add a single equality to Δ

$$\boxed{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta\delta}$$

Term stuff: Bottom, negative info, positive info + generativity, positive info + univalence

$$\frac{x \not\approx sth \in \Delta}{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta x \approx \bot} \qquad \frac{x \approx K\ \overline{y} \in \Delta}{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta x \approx \bot}$$

$$\frac{x \not\approx K \in \Delta}{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta x \approx K\ \overline{y}} \qquad \frac{x \approx K_i\ \overline{y} \in \Delta \quad i \neq j \quad K_i \text{ and } K_j \text{ generative}}{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta x \approx K_j\ \overline{z}}$$

$$\frac{x \approx K\ \overline{\tau}\ \overline{y} \in \Delta \quad \nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta\tau_i \sim \sigma_i}{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta x \approx K\ \overline{\sigma}\ \overline{z}} \qquad \frac{x \approx K\ \overline{\tau}\ \overline{y} \in \Delta \quad \nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta y_i \approx z_i}{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta x \approx K\ \overline{\sigma}\ \overline{z}}$$

Type stuff: Hand over to unspecified type oracle

$$\frac{\tau_1 \text{ and } \tau_2 \text{ incompatible to Givens in } \Delta \text{ according to type oracle}}{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta\tau_1 \sim \tau_2}$$

Mixed: Instantiate K and see if that leads to a contradiction TODO: Proper instantiation

$$\frac{\overline{\nvdash_{\textsc{Sat}} \Gamma \vdash y \triangleright \Delta \cup y \not\approx \bot}}{\nvdash_{\textsc{Sat}} \Gamma \vdash \wedge\Delta x \approx K\ \overline{y}}$$