A Categorical Perspective on Pattern Unification

Andrea Vezzosi

Joint work with Andreas Abel

Department of Computer Science and Engineering
Chalmers University of Technology and Gothenburg University, Sweden

July 13, 2014
Pattern Unification

*Pattern* unification is a subset of *Higher-Order* unification which is both **decidable** and has **most general unifiers**

- Introduced by Dale Miller in 1991 for the Simply Typed Lambda Calculus
- Still the basis of unification algorithms for type inference and proof search (Agda, Twelf, λProlog, ..)
Why *pattern*?

*pattern* = list of distinct object variables

unification problem:

\[
M \ x_0 \ x_1 \ x_2 = t
\]
Why *pattern*?

*pattern* = list of distinct object variables

unification problem: \( M \overset{\text{pattern}}{\overbrace{x_0 \ x_1 \ x_2}} = t \)

solution: \( M := \lambda x_0 \ x_1 \ x_2. \ t \)

If the assignment is well-scoped: \( \text{FV}(t) \subseteq \{x_0, x_1, x_2\} \)
Examples

Terms  \( t ::= \lambda x.t \mid tt \mid x \mid M \)

In the pattern fragment:

\[
M \, x \, y = x
\]
\[
M \, x \, y = \lambda z. \, N \, y \, z
\]

Not in the pattern fragment:

\[
M \, x \, x = x
\]
\[
M \, x \, (N \, x) = x
\]
Contextually

We reshape $\Delta \vdash M \ x_0 \ x_1 \ x_2 : \iota$

Meta as function: $M : \tau_0 \rightarrow \tau_1 \rightarrow \tau_2 \rightarrow \iota$

Meta as value in context: $\frac{y_i : \tau_i \vdash M : \iota}{\text{(}y_i \text{ fresh)}}$
Contextually

We reshape $\Delta \vdash M \ x_0 \ x_1 \ x_2 : \iota$

Meta as function: $M : \tau_0 \rightarrow \tau_1 \rightarrow \tau_2 \rightarrow \iota$

Meta as value in context: $\frac{y_i : \tau_i \vdash M : \iota}{\overrightarrow{y_i} \ fresh}$

\textit{pattern} as renaming: $\frac{\overrightarrow{y_i} : \tau_i}{p \rightarrow \Delta}$

$p = (y_0 := x_0, y_1 := x_1, y_2 := x_2)$ where $\forall i, \Delta \vdash x_i : \tau_i$
Contextually

We reshape $\Delta \vdash M \ x_0 \ x_1 \ x_2 : \iota$

Meta as function: $M : \tau_0 \to \tau_1 \to \tau_2 \to \iota$

Meta as value in context: $\frac{\tau_i}{y_i} \vdash M : \iota \quad (y_i \text{ fresh})$

**pattern as renaming:** $\frac{\tau_i}{y_i} \xrightarrow{p} \Delta$

$p = (y_0 := x_0, y_1 := x_1, y_2 := x_2) \text{ where } \forall i, \Delta \vdash x_i : \tau_i$

New form $\Delta \vdash [p] M : \iota$

$\iota \xrightarrow{M} \frac{\tau_i}{y_i} \xrightarrow{p} \Delta$
Solving the problem again

original problem: \( M x_0 x_1 x_2 = t \)

new problem: \([p] M = t\)

well-scopedness: \( \text{FV}(t) \subseteq p \iff \exists t', [p] t' = t \)

solution: \( M := t' \)
Unique solutions from Injectivity

problem: \([p] M = t\)
solution: \(M := t'\) with \([p] t' = t\), is it the most general?

Recall \(p = x_0, x_1, x_2\) with \(x_i \neq x_j \forall i \neq j\)

\(p\) injective: \([p] u = [p] u' \Rightarrow u = u'\)

\[
\exists t', [p] t' = t
\]
\[\Downarrow\]
\[
\exists! t', [p] t' = t
\]
Same Meta, Different Patterns


solution: \(M := [e] M', \text{ for some pattern } e\)

New constraint:

\[
\uparrow \\
[p \circ e] M' = [q \circ e] M' \\
\uparrow \\
p \circ e = q \circ e
\]
constraint: $p \circ e = q \circ e$

$$
\begin{align*}
\Delta_E & \overset{e}{\longrightarrow} \Delta_M & \overset{p}{\longrightarrow} \Delta \\
& \overset{q}{\longrightarrow}
\end{align*}
$$
constraint: \( p \circ e = q \circ e \)
Most Generality


Our solution: $M := [e] M'$

Alternative solution: $M := s$ with $[p] s = [q] s$

most generality constraint: $[e] M' = s$
Most Generality

problem: $\left[p\right] M := \left[q\right] M$

Our solution: $M := [e] M'$

Alternative solution: $M := s$ with $\left[p\right] s = \left[q\right] s$

most generality constraint: $[e] M' = s$

\[
\begin{array}{c}
\Delta_e \\
\uparrow \hspace{1cm} u \\
\downarrow \hspace{1cm} \delta \\
\Delta \\
\end{array}
\xrightarrow{[e]} \Delta_M \xrightarrow{[p]} [q] \Delta

M := u
Different Metas

problem: $[p] M = [q] N$

solution: $M := [r_1] M', N := [r_2] M'$
Object Variable

problem: \[[p] \, M = x\]

Two cases:

- \(\exists y, py = x\) \(\Rightarrow\) \(M := y\)
- \(\forall y, py = x\) \(\Rightarrow\) No Solution
Object Variable

problem: $[p] \, M = x$

Two cases:

1. $\exists y, py = x \Rightarrow M := y$
2. $\nexists y, py = x \Rightarrow \text{No Solution}$

The cases for abstraction and application work by congruence.
Future Work

Future work:

- Generalize to an arbitrary language where renaming has the right properties.
- Extend to dynamic pattern unification.