Relating Nominal and Higher-Order Pattern Unification

James Cheney

University of Edinburgh

UNIF 2005

April 22, 2005

Motivation

- *Higher-order unification*: studied since ca. 1970
- Undecidable, infinitary, though [Huet 1975]'s algorithm often works well in practice
- *Higher-order pattern unification* [Miller 1991]: efficiently ($O(n)$) decidable, unitary special case
- Claim: HOPU "least extension of FOU with support for name-binding"

Motivation

- *Nominal unification*: unifies terms with names and binding axiomatized using *swapping* and *freshness* [Urban, Pitts, and Gabbay 2003]

- Nice properties: $O(n^2)$, unitary

- [UPG03] observed similarities and possible reduction from NU to HOPU

Our goal

- Understand exact relationship between two approaches

- What can one do that the other cannot?

- Efficient (linear) nominal unification via HOPU?

- Semantics for higher-order patterns via nominal terms?

Higher-order patterns

- Higher-order patterns are λ-terms (with "metavariables" F, G) such that for every subterm of the form $F\,\bar{t}$, we have $\bar{t}$ a list of *distinct* bound variables.

- Yes:

$$\lambda \mathsf{x}, \mathsf{y}. F\ \mathsf{y}\ \mathsf{x} \qquad \lambda \mathsf{x}, \mathsf{y}. \mathsf{x}\ (F\ \mathsf{y}\ \mathsf{x})\ (\lambda \mathsf{z}. G\ \mathsf{z}\ \mathsf{y})$$

- No:

$$\lambda \mathsf{x}, \mathsf{y}. F\ (\mathsf{y}\ \mathsf{x}) \qquad \lambda \mathsf{x}, \mathsf{y}, \mathsf{z}. F\ (\mathsf{x}\ \mathsf{y}\ \mathsf{z})\ (G\ \mathsf{z}\ \mathsf{y})$$

Higher-order patterns

- We will use a refined language and type system for higher-order patterns.

$$
\begin{array}{rcll}
\mathsf{x}, \mathsf{y} & \in & \mathbb{A} & \text{Vars} \\
c, d & \in & CnstSym & \text{Uninterpreted constant symbols} \\
\tau & ::= & \delta \mid \tau \to \tau' & \text{types} \\
\Sigma & ::= & \cdot \mid \Sigma, c : \tau & \text{signatures} \\
\Gamma & ::= & \cdot \mid \Gamma, X : \tau & \text{contexts} \\
t, u & ::= & c \mid \mathsf{x}_\tau \mid t\ t' \mid \lambda \mathsf{x}_\tau . t & \lambda\text{-terms} \\
 & \mid & X \mid t\,\hat{}\,\mathsf{x}_\tau & \text{flexible terms}
\end{array}
$$

- Note that *bound variables are tagged with types*, whereas *metavariables are typed in* Γ.

Higher-order patterns

- Three judgments: *normal* ($\uparrow$), *rigid atomic* ($\downarrow$), and *flexible atomic* ($\Downarrow$)

$$\frac{\Gamma \vdash t \downarrow \delta}{\Gamma \vdash t \uparrow \delta} \quad \frac{\Gamma \vdash t \Downarrow \delta}{\Gamma \vdash t \uparrow \delta} \quad \frac{\Gamma \vdash t \uparrow \tau'}{\Gamma \vdash \lambda \mathsf{x}_\tau . t \uparrow \tau \to \tau'} \qquad \text{Normal}$$

$$\frac{c : \tau \in \Sigma}{\Gamma \vdash c \downarrow \tau} \quad \frac{}{\Gamma \vdash \mathsf{x}_\tau \downarrow \tau} \quad \frac{\Gamma \vdash t \downarrow \tau \to \tau' \quad \Gamma \vdash u \uparrow \tau}{\Gamma \vdash t\ u \downarrow \tau'} \qquad \text{Rigid atomic}$$

$$\frac{}{\Gamma, X : \tau \vdash X \Downarrow \tau} \quad \frac{\Gamma \vdash t \Downarrow \tau \to \tau' \quad (\mathsf{x} \notin FV(t))}{\Gamma \vdash t\,\hat{}\,\mathsf{x}_\tau \Downarrow \tau'} \qquad \text{Flexible atomic}$$

- Equational laws:

$$(\lambda \mathsf{x}.t)\ \mathsf{y} \approx_{\beta_0} t[\mathsf{y}/\mathsf{x}] \qquad t : \tau \to \tau' \approx_\eta \lambda \mathsf{x}.(t\ \mathsf{x}) \quad (\mathsf{x} \notin FV(t))$$

Higher-order patterns

- Three judgments: *normal* ($\uparrow$), *rigid atomic* ($\downarrow$), and *flexible atomic* ($\Downarrow$)

$$\frac{\Gamma \vdash t \downarrow \delta}{\Gamma \vdash t \uparrow \delta} \quad \frac{\Gamma \vdash t \Downarrow \delta}{\Gamma \vdash t \uparrow \delta} \quad \frac{\Gamma \vdash t \uparrow \tau'}{\Gamma \vdash \lambda \mathsf{x}_\tau . t \uparrow \tau \rightarrow \tau'} \qquad \text{Normal}$$

$$\frac{c : \tau \in \Sigma}{\Gamma \vdash c \downarrow \tau} \quad \frac{}{\Gamma \vdash \mathsf{x}_\tau \downarrow \tau} \quad \frac{\Gamma \vdash t \downarrow \tau \rightarrow \tau' \quad \Gamma \vdash u \uparrow \tau}{\Gamma \vdash t\ u \downarrow \tau'} \qquad \text{Rigid atomic}$$

$$\frac{}{\Gamma, X : \tau \vdash X \Downarrow \tau} \quad \frac{\Gamma \vdash t \Downarrow \tau \rightarrow \tau' \quad (\mathsf{x} \notin FV(t))}{\Gamma \vdash t \hat{}\, \mathsf{x}_\tau \Downarrow \tau'} \qquad \text{Flexible atomic}$$

- Note: pattern restriction enforced here

Nominal patterns

- We consider *nominal terms* of the following restricted form:

$$
\begin{array}{rcll}
\mathsf{a},\mathsf{b} & \in & \mathbb{A} & \text{Names} \\
c,d & \in & CnstSym & \text{Uninterpreted constant symbols} \\
\tau & ::= & \delta \mid \sigma \to \tau & \text{first-order types} \\
\sigma & ::= & \delta \mid \nu \mid \langle \nu \rangle \sigma & \text{base types} \\
\Sigma & ::= & \cdot \mid \Sigma, c : \tau & \text{signatures} \\
\Gamma & ::= & \cdot \mid \Gamma, X : \sigma & \text{contexts} \\
t & ::= & c \mid t\ t' \mid X & \text{first-order terms} \\
 & \mid & \mathsf{a}_\nu \mid \langle \mathsf{a}_\nu \rangle t \mid t \mathbin{@} \mathsf{a}_\nu & \text{nominal patterns}
\end{array}
$$

- Note that *names are tagged with name-types* ν, whereas *metavariables are assigned* σ*-types in* Γ.

Nominal patterns

- Two judgments: *normal* ($\uparrow$), *atomic* ($\downarrow$)

$$\frac{\Gamma \vdash t \downarrow \epsilon \quad (\epsilon = \delta, \nu)}{\Gamma \vdash t \uparrow \epsilon} \qquad \frac{\Gamma \vdash t \uparrow \sigma}{\Gamma \vdash \langle \mathsf{a}_\nu \rangle t \uparrow \langle \nu \rangle \sigma} \qquad \text{Normal}$$

$$\frac{c : \tau \in \Sigma}{\Gamma \vdash c \downarrow \tau} \qquad \frac{}{\Gamma \vdash \mathsf{a}_\nu \downarrow \nu} \qquad \frac{\Gamma \vdash t \downarrow \tau \to \tau' \quad \Gamma \vdash u \uparrow \tau}{\Gamma \vdash t\ u \downarrow \tau'} \qquad \text{Atomic}$$

$$\frac{}{\Gamma, X : \sigma \vdash X \downarrow \sigma} \qquad \frac{\Gamma \vdash t \downarrow \langle \nu \rangle \sigma \quad (\mathsf{a} \notin FN(t))}{\Gamma \vdash t \,@\, \mathsf{a}_\nu \downarrow \sigma} \qquad \text{Atomic}$$

- Equational laws (where $(\mathsf{a}\ \mathsf{b}) \cdot t = t[\mathsf{a}/\mathsf{b}, \mathsf{b}/\mathsf{a}]$:

$$(\langle \mathsf{a} \rangle t) \,@\, \mathsf{b} \approx_\beta (\mathsf{a}\ \mathsf{b}) \cdot t \qquad t : \langle \nu \rangle \tau \approx_\eta \langle \mathsf{a} \rangle (t \,@\, \mathsf{a})$$

Metavariables

- In HOPU, metavariables can only be replaced with *closed terms* (no free variables).
- We adopt this convention for nominal pattern unification also (no free names).
- This is not the case in ordinary NU: metavariables can be replaced with terms mentioning “free names”.
- We will return to this at the end.

Key result

- Nominal pattern unification translates to a special case of HOPU.

$$\begin{array}{rclcrcl} c^* & = & c & \quad & \mathsf{a}_\nu^* & = & \mathsf{a}_\nu \\ (t\ u)^* & = & t^*\ u^* & & (\langle \mathsf{a}_\nu \rangle t)^* & = & \lambda \mathsf{a}_\nu . t^* \\ X^* & = & X & & (t \mathbin{@} \mathsf{a}_\nu)^* & = & (t^*)\hat{\ }\mathsf{a}_\nu \end{array}$$

Lemma 1. *If* $\Gamma \vdash t : \tau$ *is a nominal pattern and* $\mathsf{b} \notin FN(t)$*, then*

$$((\mathsf{a}\ \mathsf{b}) \cdot t)^* = (t[\mathsf{a}/\mathsf{b}, \mathsf{b}/\mathsf{a}])^* = (t[\mathsf{b}/\mathsf{a}])^* = t^*[\mathsf{b}/\mathsf{a}]$$

- The translation preserves types and β and η laws and is one-to-one. So, unification can be decided by translation.

Next step

- $(-)^*$ injective & total but not surjective

- Example:

$$\lambda \mathsf{x}, \mathsf{y}.\mathsf{x}\ \mathsf{y} \approx \lambda \mathsf{x}, \mathsf{y}.F\ \mathsf{x}\ \mathsf{y}$$

  since x used as a function.

- Nevertheless, it *can* be translated to the equivalent problem

$$\langle \mathsf{x} \rangle \langle \mathsf{y} \rangle app(var(\mathsf{x}), \mathsf{y}) \approx \langle \mathsf{x} \rangle \langle \mathsf{y} \rangle F \mathbin{@} \mathsf{x} \mathbin{@} \mathsf{y}$$

Idea

- Idea: Let the base types be V_τ of "variable names of type τ", E_δ of "expressions of type δ"

- Define $E_{\tau \to \tau'} = \langle V_\tau \rangle E_{\tau'}$

- Use explicit function symbols $var : V_\tau \to E_\tau$ and $app : E_{\tau \to \tau'} \to E_\tau \to E_{\tau'}$.

- Translate as follows:

$$\begin{array}{rclcrcl}
c^{**} & = & c & \qquad & (t\ u)^{**} & = & app(t^{**}, u^{**}) \\
\mathsf{x}_\tau{}^{**} & = & var(\mathsf{x}_{\nu_\tau}) & & X^{**} & = & X \\
(\lambda \mathsf{x}.t)^{**} & = & \langle \mathsf{x} \rangle t^{**} & & (t\hat{}\mathsf{x})^{**} & = & t^{**} \,@\, \mathsf{x}
\end{array}$$

Main result

- This translation is injective and preserves types, β, and η. So any HOPU problem can be solved by translation.

  Theorem 2. *A higher-order pattern unification problem $t \approx ?\ u$ in ηlβn-normal form has a solution if and only if its translation $t^{**} \approx ?\ u^{**}$ has a nominal pattern unifier.*

Not done yet

- Nominal patterns were invented for the purpose of relating HOPU to NU.
- Still need to relate nominal patterns with "full" nominal unification.
- In particular, can NPU problems actually be translated to NU problems?
- Two problems: NU lacks concretion, and NU unifiers can substitute open terms for metavariables

Nominal terms

$$\begin{array}{rcll}
\mathsf{a}, \mathsf{b} & \in & \mathbb{A} & \text{Names} \\
c, d & \in & CnstSym & \text{Uninterpreted constant symbols} \\
\Sigma & ::= & \cdot \mid \Sigma, c : \tau & \text{signatures} \\
\Gamma & ::= & \cdot \mid \Gamma, X : \sigma & \text{contexts} \\
t, u & ::= & X \mid c \mid t\ u & \text{First-order terms (applicative style)} \\
 & \mid & \langle \mathsf{a} \rangle t \mid (\mathsf{a}\ \mathsf{b}) \cdot t \mid \mathsf{a} & \text{Nominal terms} \\
C & ::= & t \approx u \mid \mathsf{a} \# t & \text{Equality, freshness constraints}
\end{array}$$

Note: metavariables can mention free names!

Ground swapping

The result of applying a swapping permutation to a ground term is:

$$\begin{array}{rcl} (\mathsf{a}\ \mathsf{b}) \cdot \mathsf{c} & = & \begin{cases} \mathsf{b} & (\mathsf{a} = \mathsf{c}) \\ \mathsf{a} & (\mathsf{b} = \mathsf{c}) \\ \mathsf{c} & (\mathsf{a} \neq \mathsf{c} \neq \mathsf{b}) \end{cases} \\ (\mathsf{a}\ \mathsf{b}) \cdot c & = & c \\ (\mathsf{a}\ \mathsf{b}) \cdot (t\ u) & = & ((\mathsf{a}\ \mathsf{b}) \cdot t)\ ((\mathsf{a}\ \mathsf{b}) \cdot u) \\ (\mathsf{a}\ \mathsf{b}) \cdot \langle \mathsf{c} \rangle t & = & \langle (\mathsf{a}\ \mathsf{b}) \cdot \mathsf{c} \rangle (\mathsf{a}\ \mathsf{b}) \cdot t \end{array}$$

For nominal terms, permutations applied to metavariables are “suspended” (since metavariables can mention names).

Ground freshness theory

$$\frac{(\mathsf{a} \neq \mathsf{b})}{\mathsf{a} \,\#\, \mathsf{b}}$$ Different names fresh

$$\frac{}{\mathsf{a} \,\#\, c}$$ Anything fresh for constant

$$\frac{\mathsf{a} \,\#\, t \quad \mathsf{a} \,\#\, u}{\mathsf{a} \,\#\, t\ u}$$ Freshness ignores function application

$$\frac{}{\mathsf{a} \,\#\, \langle \mathsf{a} \rangle t}$$ Fresh if bound

$$\frac{(\mathsf{a} \neq \mathsf{b}) \quad \mathsf{a} \,\#\, t}{\mathsf{a} \,\#\, \langle \mathsf{b} \rangle t}$$ Fresh if fresh for body

Ground equational theory

$$\left.\begin{array}{c}
\dfrac{}{\mathsf{a} \approx \mathsf{a}} \\[2ex]
\dfrac{}{c \approx c} \\[2ex]
\dfrac{t_1 \approx u_1 \quad t_2 \approx u_2}{t_1\ t_2 \approx u_1\ u_2} \\[2ex]
\dfrac{t \approx u}{\langle \mathsf{a} \rangle t \approx \langle \mathsf{a} \rangle u}
\end{array}\right\} \quad \text{Standard equational rules}$$

$$\dfrac{(\mathsf{a} \neq \mathsf{b}) \quad \mathsf{a} \# u \quad t \approx (\mathsf{a}\ \mathsf{b}) \cdot u}{\langle \mathsf{a} \rangle t \approx \langle \mathsf{b} \rangle u} \quad \alpha\text{-equivalence for abstractions}$$

Solving the first problem

- We can translate out concretion using the following property:

$$\langle \mathsf{a} \rangle t \approx u \iff t \approx u \mathbin{@} \mathsf{a}$$

- This works *only* if $\mathsf{a} \# u$, that is, $u \mathbin{@} \mathsf{a}$ is well-formed.

- Thus, we can remove concretion by translating:

$$P[t \mathbin{@} \mathsf{a}] \iff \exists X. P[X] \wedge \langle \mathsf{a} \rangle X \approx t$$

- Note that X may mention a.

Solving the second problem

- We need to translate nominal unifiers $\langle\nabla, \theta\rangle$ (θ open) to nominal pattern unifiers θ' (closed)

- This is tricky; I'll show an example and gloss over details.

- Also need to be careful about empty types, but this is a standard problem.

An example

- Given

$$\langle \mathsf{a}\rangle\langle \mathsf{b}\rangle X \mathbin{@} \mathsf{b} \mathbin{@} \mathsf{a} \approx \langle \mathsf{a}\rangle\langle \mathsf{b}\rangle Y \mathbin{@} \mathsf{a}$$

- we assume $a, b \# X, Y$ and substitute

$$X = \langle \mathsf{b}\rangle\langle \mathsf{a}\rangle X', Y = \langle \mathsf{a}\rangle Y'$$

- This gives us a NU problem

$$\{\mathsf{a}, \mathsf{b}\} \# \{\langle \mathsf{b}\rangle\langle \mathsf{a}\rangle X', \langle \mathsf{a}\rangle Y'\}, \langle \mathsf{a}\rangle\langle \mathsf{b}\rangle X' \approx \langle \mathsf{a}\rangle\langle b\rangle Y'$$

- with solution $\langle \{b \# Y'\}, [X' = Y']\rangle$

An example

- We have:

$$X = \langle \mathsf{b} \rangle \langle \mathsf{a} \rangle Y', Y = \langle \mathsf{a} \rangle Y', \mathsf{b} \# Y'$$

- Now we want to solve for X, Y in terms of closed metavariables.

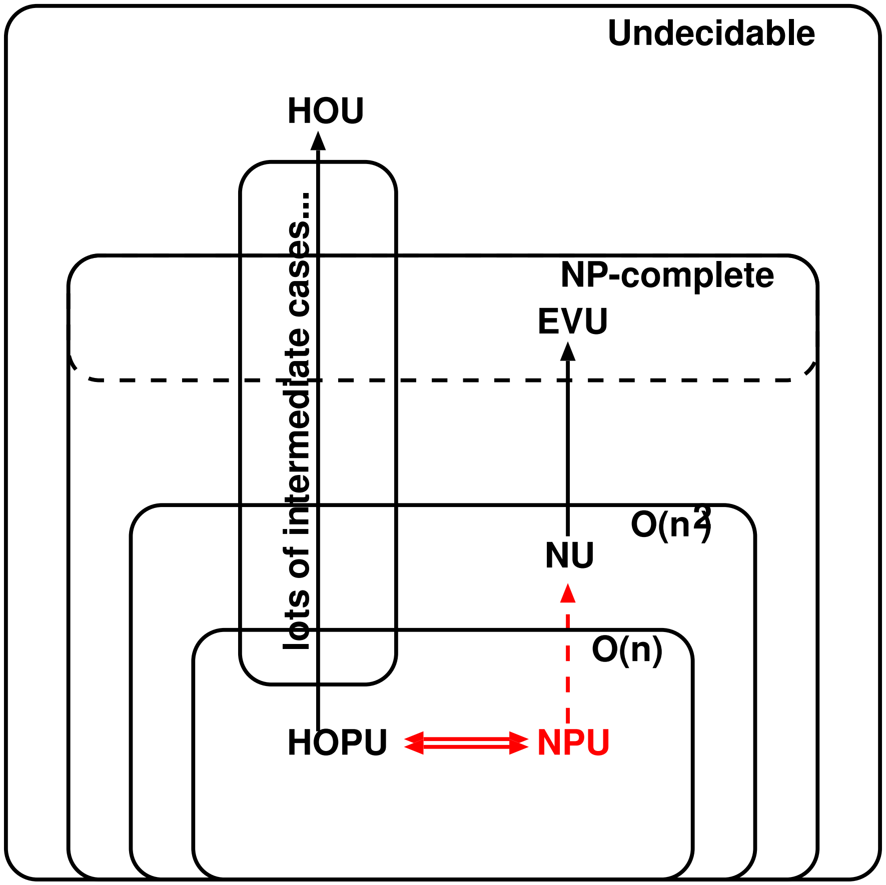
- Since $b \# Y'$, substitute $Y' = Z \mathbin{@} \mathsf{a}$ to obtain

$$X = \langle \mathsf{b} \rangle \langle \mathsf{a} \rangle Z \mathbin{@} \mathsf{a}, Y = \langle \mathsf{a} \rangle Z \mathbin{@} \mathsf{a}$$

- which is the most general solution:

$$\langle \mathsf{a} \rangle \langle \mathsf{b} \rangle (\langle \mathsf{b} \rangle \langle \mathsf{a} \rangle Z \mathbin{@} \mathsf{a}) \mathbin{@} \mathsf{b} \mathbin{@} \mathsf{a} \approx \langle \mathsf{a} \rangle \langle \mathsf{b} \rangle Z \mathbin{@} \mathsf{a} \approx \langle \mathsf{a} \rangle \langle \mathsf{b} \rangle (\langle \mathsf{a} \rangle Z \mathbin{@} \mathsf{a}) \mathbin{@} \mathsf{a}$$

Big picture



Undecidable
HOU
NP-complete
EVU
lots of intermediate cases...
O(n2)
NU
O(n)
HOPU
NPU

Related work

- [Miller 91] showed that full HOU could be translated to L_λ programs

- [Hamana 2001,2002] studied unification/LP for binding algebra terms, similar but slightly less restricted than patterns. Apparently NP, exact complexity unknown

- [Urban et al 2004] discuss reducing NU to HOPU; seems much harder to translate answers back

Future work

- Translating L_λ to αProlog, $FO\lambda^\nabla$ to NL?
- Exact complexity bounds for reductions, nominal unification? (better than $O(n^2)$?)
- HOU, β_0 unification, π-calculus structural congruence unification as nominal equational unification?

Conclusion

- Showed that HOPU can be simulated by NU via a straightforward translation.
- Reverse direction (HOPU to NU), exact complexity of NU still unclear.
- Intermediate NPU case seems interesting in its own right
- and provides an independent explanation for the pattern restriction.