Generic pattern unification

A categorical approach

Ambroise Lafont   Neel Krishnaswami

University of Cambridge

October 2022

What is unification?

$$\underbrace{t}\overset{?}{=}\underbrace{u}$$

terms with metavariables $M, N, \ldots$

Unifier = metavariable substitution σ s.t.

$$t[\sigma] = u[\sigma]$$

Most general unifier = unifier σ that uniquely factors any other

$$\forall \delta, \qquad t[\delta] = u[\delta] \quad \Leftrightarrow \quad \exists! \delta'. \ \delta = \delta' \circ \sigma$$

Goal of unification = find the most general unifier

A. Lafont, N. Krishnaswami Generic pattern unification

Where is unification used?

First-order unification

No metavariable argument

Examples

- Logic programming (Prolog)
- ML type inference systems

$(M \to N) \stackrel{?}{=} (\mathbb{N} \to M)$

Second-order unification

$M(\dots)$

Example

- Type theory, proof assistants

$(\forall x.M(x, u)) \stackrel{?}{=} t$

Undecidable

A. Lafont, N. Krishnaswami   Generic pattern unification

Pattern unification [Miller '91]

A decidable fragment of second-order unification.

Pattern restriction:

$$M(\underbrace{x_1, \ldots, x_n}_{\text{distinct variables}})$$

$\exists$ unification algorithm [Miller '91]

- fails if no unifier
- returns the most general unifier

This work

A **generic** algorithm for pattern unification

- Parameterised by a *signature*
- Categorical semantics

Examples

- *binding signatures*
- Linear syntax (e.g., quantum λ-calculus)
- Intrinsic system F

A. Lafont, N. Krishnaswami   Generic pattern unification

Related work: algebraic accounts of unification

First-order unification

- Lattice theory [Plotkin '70]
- Category theory
  - [Rydeheard-Burstall '88]
  - [Goguen '89]

Pattern unification

- Category theory
  - [Vezzosi-Abel '14]
    normalised λ-terms
  - This work

A. Lafont, N. Krishnaswami   Generic pattern unification

Outline

A. Lafont, N. Krishnaswami   Generic pattern unification

Outline

A. Lafont, N. Krishnaswami   Generic pattern unification

Outline

A. Lafont, N. Krishnaswami   Generic pattern unification

Syntax (De Bruijn levels)

$$\overbrace{\Gamma}^{\text{Metavariable context } (M_1 : n_1, \dots)} ; \underbrace{n}_{\text{Variable context}} \vdash t$$

$$\frac{x < n}{\Gamma; n \vdash x} \text{Var} \qquad \frac{\Gamma; n \vdash t \quad \Gamma; n \vdash u}{\Gamma; n \vdash t \; u} \text{App} \qquad \frac{\Gamma; n + 1 \vdash t}{\Gamma; n \vdash \lambda t} \text{Abs}$$

$$\frac{(M : n) \in \Gamma \quad x_1, \dots, x_n < n \quad x_1, \dots x_n \text{ distinct}}{\Gamma; n \vdash M(x_1, \dots, x_n)} \text{Flex}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Metavariable substitution

Substitution σ from $\overbrace{(M_1 : m_1, \ldots, M_p : m_p)}^{\Gamma}$ **to** Δ:

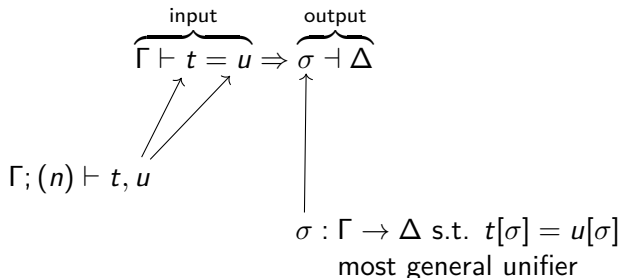$$(\sigma_1, \ldots, \sigma_p) \qquad \text{s.t.} \qquad \Delta; m_i \vdash \sigma_i$$

σ extends to terms:

$$\Gamma; n \vdash t \quad \mapsto \quad \Delta; n \vdash t[\sigma]$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Outline

A. Lafont, N. Krishnaswami   Generic pattern unification

Unification algorithm

$$\overbrace{\Gamma \vdash t = u}^{\text{input}} \Rightarrow \overbrace{\sigma \dashv \Delta}^{\text{output}}$$

$$\Gamma; (n) \vdash t, u$$

$$\sigma : \Gamma \to \Delta \text{ s.t. } t[\sigma] = u[\sigma]$$
$$\text{most general unifier}$$

Examples
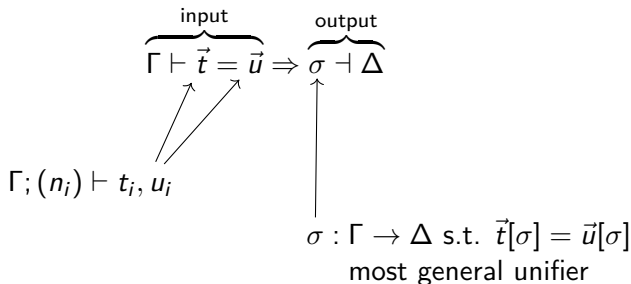
$$\Gamma, M : 2 \vdash M(5,3) = 5 \Rightarrow (M \mapsto 0) \dashv \Gamma$$

$$\Gamma, M : 2 \vdash M(5,3) = 3 \Rightarrow (M \mapsto 1) \dashv \Gamma$$

$$\frac{\Gamma \vdash t = u \Rightarrow \sigma \dashv \Delta}{\Gamma \vdash \lambda t = \lambda u \Rightarrow \sigma \dashv \Delta}$$

$$\frac{\Gamma \vdash "t_1, t_2 = u_1, u_2" \Rightarrow \sigma \dashv \Delta}{\Gamma \vdash t_1 \ t_2 = u_1 \ u_2 \Rightarrow \sigma \dashv \Delta}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Unifying lists of terms

$$\overbrace{\Gamma \vdash \vec{t} = \vec{u}}^{\text{input}} \Rightarrow \overbrace{\sigma \dashv \Delta}^{\text{output}}$$

$$\Gamma \, ; (n_i) \vdash t_i, u_i$$

$$\sigma : \Gamma \to \Delta \text{ s.t. } \vec{t}[\sigma] = \vec{u}[\sigma]$$
$$\text{most general unifier}$$

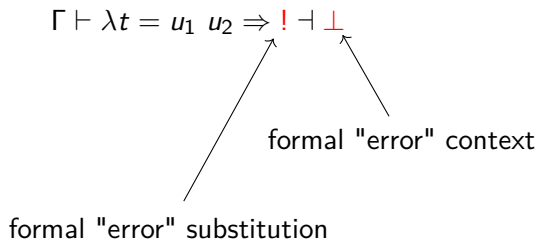A. Lafont, N. Krishnaswami   Generic pattern unification

Examples (lists)

$$\Gamma \vdash () = () \Rightarrow id_\Gamma \dashv \Gamma$$

$$\frac{\Gamma \vdash t_1 = u_1 \Rightarrow \sigma_1 \dashv \Delta_1 \qquad \Delta_1 \vdash \vec{t_2}[\sigma_1] = \vec{u_2}[\sigma_1] \Rightarrow \sigma_2 \dashv \Delta_2}{\Gamma \vdash t_1, \vec{t_2} = u_1, \vec{u_2} \Rightarrow \sigma_1[\sigma_2] \dashv \Delta_2} \text{U-SPLIT}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Impossible cases

$$\Gamma \vdash \lambda t = u_1 \ u_2 \Rightarrow \ ! \dashv \perp$$

formal "error" context

formal "error" substitution

Unifying a metavariable $M(\vec{x}) \stackrel{?}{=} \ldots$

Three cases

1. $M(\vec{x}) \stackrel{?}{=} M(\vec{y})$

2. $M(\vec{x}) \stackrel{?}{=} \ldots M(\vec{y}) \ldots$

3. $M(\vec{x}) \stackrel{?}{=} u$ and $M \notin u$

A. Lafont, N. Krishnaswami   Generic pattern unification

Unifying a metavariable with itself

$$M(\vec{x}) \stackrel{?}{=} M(\vec{y})$$

Most general unifier: $M \mapsto M'(\vec{z})$

- $\vec{z} = $ vector of common positions: $x_{\vec{z}} = y_{\vec{z}}$

Formally,

$$\frac{"n \vdash \vec{x} = \vec{y} \Rightarrow \vec{z} \dashv p"}{\Gamma, M : n \vdash M(\vec{x}) = M(\vec{y}) \Rightarrow M \mapsto M'(\vec{z}) \dashv \Gamma, M' : p}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Cyclic case

$$M(\vec{x}) \stackrel{?}{=} \dots M(\vec{y}) \dots$$

No unifier

$$\Gamma \vdash \underbrace{M(\vec{x})}_{} = \underbrace{\dots M(\vec{y}) \dots}_{} \Rightarrow \; ! \dashv \bot$$

sizes cannot match after substitution

A. Lafont, N. Krishnaswami   Generic pattern unification

Non cyclic case

$$M(\vec{x}) \stackrel{?}{=} u \ (M \notin u)$$

Most general unifier: $M \mapsto u[\vec{x}^{-1}]$

- Requires

$$fv(u) \subset \vec{x} \tag{1}$$

$\Rightarrow$ **Pruning phase**: enforces (1) by restricting metavariable arities.

Example

$$
\begin{array}{ccc}
M(x) & \stackrel{?}{=} & N(x, y) \\
N(x, y) & \xrightarrow{\text{pruning}} & N'(x)
\end{array}
$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Pruning phase

u after pruning and renamed by $\vec{x}^{-1}$

$$\Gamma \vdash u :> M(\vec{x}) \Rightarrow \overbrace{v} \; ; \sigma \dashv \Delta$$

$M \notin \Gamma, u$

$\sigma : \Gamma \to \Delta$

pruning substitution

Intuition

$$u \stackrel{?}{=} M(\vec{x}) \quad \Rightarrow \quad (\sigma, M \mapsto v) = \text{most general unifier}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Pruning a metavariable

$$M(\vec{x}) \stackrel{?}{=} N(\vec{y})$$

Most general unifier: $M \mapsto N'(\vec{l})$, $N \mapsto N'(\vec{r})$ such that

$$x_{\vec{l}} = y_{\vec{r}}$$

$$\frac{"n \vdash \vec{x} :> \vec{y} \Rightarrow \vec{l}; \vec{r} \dashv p"}{\Gamma, N : n \vdash N(\vec{x}) :> M(\vec{y}) \Rightarrow N'(\vec{l}); N \mapsto N'(\vec{r}) \dashv \Gamma, N' : p}$$

23/55

Pruning: other examples

$$\overline{\Gamma \vdash x_i :> M(x_0, \ldots, x_n) \Rightarrow i; id_\Gamma \dashv \Gamma}$$

$$\frac{y \notin \vec{x}}{\Gamma \vdash y :> M(\vec{x}) \Rightarrow !; ! \dashv \bot}$$

$$\frac{\overset{\text{bound variable}}{\Gamma \vdash t :> M_1(\vec{x}, \overbrace{n}) \Rightarrow v; \sigma \dashv \Delta}}{\Gamma \vdash \lambda t :> M(\vec{x}) \Rightarrow \lambda v; \sigma \dashv \Delta}$$

$$\frac{"\Gamma \vdash t, u :> M_1(\vec{x}), M_2(\vec{x}) \Rightarrow v_1, v_2; \sigma \dashv \Delta"}{\Gamma \vdash t\ u :> M(\vec{x}) \Rightarrow v_1\ v_2; \sigma \dashv \Delta}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Pruning multi-terms

$$\Gamma \vdash u_1, \ldots, u_n :> M_1(\vec{x}_1), \ldots M_n(\vec{x}_n) \Rightarrow v_1, \ldots, v_n; \sigma \dashv \Delta$$

$$\Gamma; (n_i) \vdash u_i \qquad\qquad \Delta; m_i \vdash v_i$$

$$(M_i : m_i) \notin \Gamma, u_j \qquad\qquad \sigma : \Gamma \to \Delta$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Rules for multi-terms

$$\overline{\Gamma \vdash () :> () \Rightarrow (); id_\Gamma \dashv \Gamma}$$

$$\frac{\Gamma \vdash t_1 :> M_1(\vec{x}) \Rightarrow u_1; \sigma_1 \dashv \Delta_1 \quad \Delta_1 \vdash \vec{t_2}[\sigma_1] :> \vec{M_2} \Rightarrow \vec{u_2}; \sigma_2 \dashv \Delta_2}{\Gamma \vdash t_1, \vec{t_2} :> M_1(\vec{x}), \vec{M_2} \Rightarrow u_1[\sigma_2], \vec{u_2}; \sigma_1[\sigma_2] \dashv \Delta_2}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Outline

Parameterisation by a *signature*

Binding signature for pure λ-calculus

$$app : (0, 0) \qquad abs : (1)$$

number of bound variables in the argument

Example: pruning an operation

$$o : (\alpha_1, \ldots, \alpha_p)$$

$$\frac{\Gamma \vdash \vec{t} :> M_1(\vec{x}, \overbrace{n, \ldots, n + \alpha_1 - 1}^{\text{bound variables}}), \ldots, M_p(\ldots) \Rightarrow \vec{u}; \sigma \dashv \Delta}{\Gamma \vdash o(\vec{t}) :> N(\vec{x}) \Rightarrow o(\vec{u}); \sigma \dashv \Delta}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Outline

1. Pattern unification for pure λ-calculus
   - Syntax
   - Unification algorithm

2. Generalisation to binding signatures

3. Categorical generalisation
   - A case study: syntax of pure λ-calculus
   - Generic pattern unification

4. Example: System F

A. Lafont, N. Krishnaswami   Generic pattern unification

Outline

A. Lafont, N. Krishnaswami   Generic pattern unification

Pure λ-calculus as a functor

category of finite cardinals and injections between them

Pure λ-calculus as a functor $\Lambda : \mathbb{F}_m \to \mathrm{Set}$

$$\Lambda_n = \{t \mid \cdot; n \vdash t\}$$

Pure λ-calculus as a fixpoint

$$\Lambda_n \cong \underbrace{\{0, \ldots, n-1\}}_{\text{variables}} + \underbrace{\Lambda_n \times \Lambda_n}_{\text{application}} + \underbrace{\Lambda_{n+1}}_{\text{abstraction}}$$

In fact,

$$\Lambda = \mu X.F(X)$$

Initial algebra of the endofunctor F on $[\mathbb{F}_m, \mathrm{Set}]$

$$F(X)_n = \{0, \ldots, n-1\} + X_n \times X_n + X_{n+1}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Pure λ-calculus extended with a metavariable $M : m$

$$\Lambda(m)_n = \{t \mid M : m; n \vdash t\}$$

As a fixpoint:

$$\Lambda(m) = \mu X.(\underbrace{F(X)}_{\text{operations / variables}} + arg^M)$$

$$\begin{aligned}
arg^M{}_{\boldsymbol{n}} &= \{M\text{-arguments in the variable context } \boldsymbol{n}\} \\
&= \{\vec{x} \in \{0, \ldots, n-1\}^m \mid x_1, \ldots, x_m \text{ distinct}\} \\
&= \hom_{\mathbb{F}_m}(m, n)
\end{aligned}$$

$$\Lambda(m) = \mu X.(F(X) + ym)$$

Pure λ-calculus with metavariables

$$\Lambda(\Gamma)_n = \{t \mid \Gamma; n \vdash t\}$$

As a fixpoint:

$$\Lambda(\Gamma) = \mu X.(F(X) + \underbrace{\coprod_{(M:m) \in \Gamma} ym}_{\underline{\Gamma}})$$

$$= \underbrace{T}_{\text{free monad generated by } F}(\underline{\Gamma})$$

$$T(\underline{\Gamma})_n = \{t \mid \Gamma; n \vdash t\}$$

Unification as a Kleisli coequaliser

Claims:

- $\hom(yn, T\underline{\Gamma})$ = set of terms in context $\Gamma; n$.
- $\hom(\underline{\Gamma}, T\underline{\Delta})$ = set of metavariable substitutions $\Gamma \to \Delta$.
- Most general unifier of t, u: coequaliser of $yn \underset{u}{\overset{t}{\rightrightarrows}} T\underline{\Gamma}$

    in $\mathsf{Th}(F) \subset \mathsf{Kl}(T)$.

Objects: $\underline{\Gamma}, \underline{\Delta}, \ldots$
(finite coproducts of representable functors)
"Lawvere theory of T"

A. Lafont, N. Krishnaswami   Generic pattern unification

Outline

A. Lafont, N. Krishnaswami   Generic pattern unification

Signature

1. $\mathcal{A}$ small category (e.g., $\mathbb{F}_m$, 1)

Intuition: objects = metavariable arities,

morphisms = metavariable arguments.

- All morphisms in $\mathcal{A}$ are monic (pattern restriction).
- $\mathcal{A}$ has finite connected limits ($M(\vec{x}) \overset{?}{=} N(\vec{y})$).

2. F endofunctor on $[\mathcal{A}, \mathrm{Set}]$ of the shape

$$F(X)_a = \coprod_{o \in O_a} X_{L_{o,1}} \times \cdots \times X_{L_{o,n_o}}$$

such that F restricts to an endofunctor on functors preserving finite connected limits.

A. Lafont, N. Krishnaswami   Generic pattern unification

Typing rules

Notation

$$\underbrace{\Gamma}_{M_1 : a_1, \ldots, M_n : a_n}; b \vdash u \qquad \text{means} \qquad u \in T(\underbrace{\Gamma}_{ya_1 + \cdots + ya_n})_b$$

$$F(X)_a = \coprod_{o \in O_a} X_{L_{o,1}} \times \cdots \times X_{L_{o,n_o}}$$

$$\frac{\Gamma; L_{o,i} \vdash t_i}{\Gamma; a \vdash o(\vec{t})}\mathrm{RIGID} \qquad \frac{x \in \hom_{\mathcal{A}}(a, b)}{\Gamma, M : a \quad ; \quad b \vdash M(x)}\mathrm{FLEX}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Semantics of unification

Claim: Given a signature $(\mathcal{A}, F)$, a coequaliser diagram in $\mathrm{Th}(F)$ has a colimit as soon as there exists a cocone (i.e., a 'unifier').

Proof: By describing a unification algorithm.

End of this section: soundness proofs for 3 rules.

A. Lafont, N. Krishnaswami   Generic pattern unification

Interpreting the unification statements

Notations

$$\Gamma \vdash t = u \Rightarrow \sigma \dashv \Delta \quad \Leftrightarrow \quad \cdot \overset{t}{\underset{u}{\rightrightarrows}} \Gamma \overset{\sigma}{-\!\!\!\rightarrow} \Delta \qquad \text{coequaliser}$$
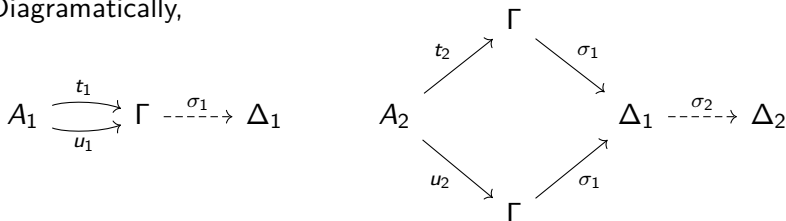
$$\Gamma \vdash t :> f \Rightarrow v; \sigma \dashv \Delta \quad \Leftrightarrow \quad \begin{array}{ccc} \cdot & \overset{f}{\longrightarrow} & \cdot \\ {\scriptstyle t}\downarrow & & \downarrow{\scriptstyle v} \\ \Gamma & \underset{\sigma}{-\!\!\!\rightarrow} & \Delta \end{array} \qquad \text{pushout}$$

mostly used in $\text{Th}(F)_\perp = \text{Th}(F) +$ a free terminal object $\perp$.

Soundness of U-SPLIT [Rydeheard-Burstall '88]

$$\frac{\Gamma \vdash t_1 = u_1 \Rightarrow \sigma_1 \dashv \Delta_1 \qquad \Delta_1 \vdash t_2[\sigma_1] = u_2[\sigma_1] \Rightarrow \sigma_2 \dashv \Delta_2}{\Gamma \vdash t_1, t_2 = u_1, u_2 \Rightarrow \sigma_1[\sigma_2] \dashv \Delta_2} \text{U-SPLIT}$$

Diagramatically,

$$A_1 \underset{u_1}{\overset{t_1}{\rightrightarrows}} \Gamma \dashrightarrow^{\sigma_1} \Delta_1$$

$$A_1 + A_2 \underset{u_1, u_2}{\overset{t_1, t_2}{\rightrightarrows}} \Gamma \dashrightarrow^{\sigma_2 \circ \sigma_1} \Delta_2$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Soundness of U-FLEXFLEX

$$\frac{b \vdash x =_{\mathcal{A}^{op}} y \Rightarrow z \dashv c}{M : b \vdash M(x) = M(y) \Rightarrow M \mapsto M'(z) \dashv M' : c} \text{U-FLEXFLEX}$$

Diagrammatically,

$$\frac{a \xrightarrow[y]{x} b - \overset{z}{-} \succ c \quad \text{in } \mathcal{A}^{op}}{\mathcal{L}a \xrightarrow[\mathcal{L}y]{\mathcal{L}x} \mathcal{L}b - \overset{\mathcal{L}z}{-} \succ \mathcal{L}c \ \text{in } \mathsf{Th}(F)}$$

where

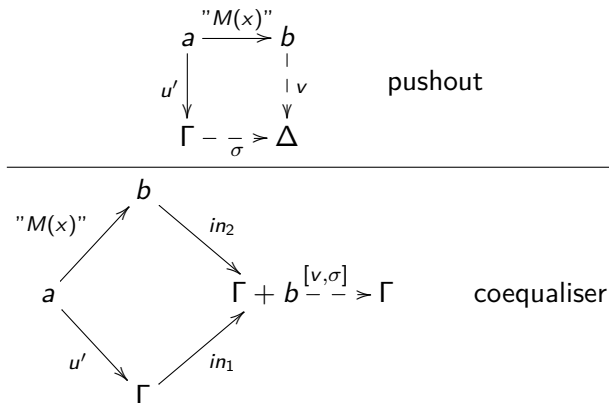$$a \xrightarrow{x} b \quad \xmapsto{\quad \mathcal{L} : \mathcal{A}^{op} \to \mathsf{Th}(F) \quad} \quad ya \xrightarrow{"M(x)"} T(\underline{M : b})$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Soundness of U-NoCycle

$$\frac{u_{|\Gamma} = \underline{u'} \qquad \Gamma \vdash u' :> M(x) \Rightarrow v; \sigma \dashv \Delta}{\Gamma, M : b \vdash M(x) = u \Rightarrow \sigma, M \mapsto v \dashv \Delta} \text{U-NoCycle}$$

Diagrammatically,

A. Lafont, N. Krishnaswami   Generic pattern unification

Outline

A. Lafont, N. Krishnaswami   Generic pattern unification

Types

Notation

$n \vdash \tau$ type $\qquad \Leftrightarrow \qquad$ the type τ is wellformed in context n

$$\frac{\alpha < n}{n \vdash \alpha \text{ type}} \text{Type-Var} \qquad \frac{n+1 \vdash \tau \text{ type}}{n \vdash \forall \tau \text{ type}} \text{Forall}$$

$$\frac{n \vdash \tau_1, \tau_2 \text{ type}}{n \vdash \tau_1 \to \tau_2 \text{ type}} \text{Arrow}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Metavariable arities

Metavariable application

$$M(\overbrace{\alpha_1, \ldots, \alpha_p}^{\text{type variables}} | \overbrace{x_1, \ldots, x_q}^{\text{"ground" variables}})$$

$$M : (p | \overbrace{\underbrace{\tau_1, \ldots, \tau_q}_{\text{input argument types}} \vdash \tau_f}^{p \vdash \tau_i \text{ type}})$$

number of type variable arguments

output type

A. Lafont, N. Krishnaswami   Generic pattern unification

Typing rule for metavariables

Typing judgement

$$\underbrace{\Gamma}_{\text{Metavariable context}} ; \overbrace{n}^{\text{Type variable context}} \mid \underbrace{\overbrace{t_1, \ldots, t_m}^{\text{Types of variables}}}_{n \vdash t_i \text{ type}} \vdash u : t_f$$

Typing metavariables

$$\frac{\alpha_1, \ldots, \alpha_p \text{ distinct}, < n \qquad x_1, \ldots x_q \text{ distinct}, < m \qquad \tau_i[\vec{\alpha}] = t_{x_i}}{\Gamma, M : (p | \tau_1, \ldots, \tau_q \vdash \tau_f) ; n \mid t_1, \ldots, t_m \vdash M(\underbrace{\vec{\alpha}}_{\text{type variables}} | \vec{x}) : \tau_f[\vec{\alpha}]}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Signature

- Objects of $\mathcal{A}$ = metavariables arities

$$n| \underbrace{\tau_1, \ldots, \tau_p}_{C} \vdash \tau_f$$

- Need an endofunctor F on $[\mathcal{A}, \mathrm{Set}]$ s.t.

$$\mu F(n|C \vdash \tau_f) = \{t \text{ s.t. } n|C \vdash t : \tau_f\}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

The endofunctor for System *F*

Typing rule	$F(X)_{n\mid C\vdash\tau} = \coprod\ldots$
$\dfrac{C_i = \tau}{n\mid C \vdash i : \tau}\text{V{\scriptsize AR}}$	$\lvert C\rvert_\tau$
$\dfrac{n\mid C, \tau_1 \vdash t : \tau_2}{n\mid C \vdash \lambda t : \tau_1 \to \tau_2}\text{A{\scriptsize BS}}$	$\coprod\limits_{\tau_1,\tau_2 \text{ s.t. } \tau = (\tau_1 \to \tau_2)} X_{n\mid C,\tau_1\vdash\tau_2}$
$\dfrac{n\mid C \vdash t : \tau' \to \tau \quad n\mid C \vdash u : \tau'}{n\mid C \vdash t\ u : \tau}\text{A{\scriptsize PP}}$	$\coprod\limits_{\tau'} X_{n\mid C\vdash\tau'\to\tau} \times X_{n\mid C\vdash\tau'}$
$\dfrac{n+1\mid C \vdash t : \tau'}{n\mid C \vdash \Lambda t : \forall\tau'}\text{T-A{\scriptsize BS}}$	$\coprod\limits_{\tau' \text{ s.t. } \tau=\forall\tau'} X_{n+1\mid C\vdash\tau'}$
$\dfrac{n\mid C \vdash t : \forall\tau_1}{n\mid C \vdash t \cdot \tau_2 : \tau_1[\tau_2]}\text{T-A{\scriptsize PP}}$	$\coprod\limits_{\tau_1,\tau_2 \text{ s.t. } \tau=\tau_1[\tau_2]} X_{n\mid C\vdash\forall\tau_1}$

Unification in system F: an example

$$M(\vec{\alpha}|\vec{x}) \stackrel{?}{=} M(\vec{\beta}|\vec{y})$$

Most general unifier: $M \mapsto N(\vec{\gamma}, \vec{z})$, where

- $\vec{\gamma}$ maximal s.t.

$$\alpha_{\vec{\gamma}} = \beta_{\vec{\gamma}}$$

- $\vec{z}$ maximal s.t.

$$x_{\vec{z}} = y_{\vec{z}}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Summary of the generic unification algorithm

$$\overline{\Gamma \vdash () = () \Rightarrow id_\Gamma \dashv \Gamma} \quad \overline{\bot \vdash \vec{t} = \vec{u} \Rightarrow \,! \dashv \bot}$$

$$\frac{\Gamma \vdash t_1 = u_1 \Rightarrow \sigma_1 \dashv \Delta_1 \quad \Delta_1 \vdash \vec{t_2}[\sigma_1] = \vec{u_2}[\sigma_1] \Rightarrow \sigma_2 \dashv \Delta_2}{\Gamma \vdash t_1, \vec{t_2} = u_1, \vec{u_2} \Rightarrow \sigma_1[\sigma_2] \dashv \Delta_2} \text{U-Split}$$

$$\frac{\Gamma \vdash \vec{t} = \vec{u} \Rightarrow \sigma \dashv \Delta}{\Gamma \vdash o(\vec{t}) = o(\vec{u}) \Rightarrow \sigma \dashv \Delta} \text{U-RigRig} \quad \frac{o \neq o'}{\Gamma \vdash o(\vec{t}) = o'(\vec{u}) \Rightarrow \,! \dashv \bot}$$

$$\frac{u_{|\Gamma} = \underline{u'} \quad \Gamma \vdash u' :> M(x) \Rightarrow v; \sigma \dashv \Delta}{\Gamma, M : b \vdash M(x) = u \Rightarrow \sigma, M \mapsto v \dashv \Delta} \text{U-NoCycle} \quad + \text{sym}$$

$$\frac{b \vdash x =_{\mathcal{A}^{op}} y \Rightarrow z \dashv c}{\Gamma, M : b \vdash M(x) = M(y) \Rightarrow M \mapsto M'(z) \dashv \Gamma, M' : c} \text{U-FlexFlex}$$

$$\frac{u = o(\vec{t}) \quad u_{|\Gamma} \neq \underline{\ldots}}{\Gamma, M : b \vdash M(x) = u \Rightarrow \,! \dashv \bot} \text{U-Cyclic} \quad + \text{sym}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Pruning phase

$$\overline{\Gamma \vdash () :> () \Rightarrow (); id_\Gamma \dashv \Gamma} \quad \overline{\bot \vdash \vec{t} :> \vec{f} \Rightarrow !; ! \dashv \bot}$$

$$\frac{\Gamma \vdash t_1 :> M_1(\vec{x}) \Rightarrow u_1; \sigma_1 \dashv \Delta_1 \quad \Delta_1 \vdash \vec{t_2}[\sigma_1] :> \vec{M_2} \Rightarrow \vec{u_2}; \sigma_2 \dashv \Delta_2}{\Gamma \vdash t_1, \vec{t_2} :> M_1(\vec{x}), \vec{M_2} \Rightarrow u_1[\sigma_2], \vec{u_2}; \sigma_1[\sigma_2] \dashv \Delta_2}$$

$$\frac{c \vdash_{\mathcal{A}^{op}} y :> x \Rightarrow l; r \dashv d}{\Gamma, M : c \vdash M(y) :> N(x) \Rightarrow M'(l); M \mapsto M'(r) \dashv \Gamma, M' : d}\text{P-Flex}$$

$$\left.\begin{array}{c} \dfrac{\Gamma \vdash \vec{t} :> \mathcal{L}^+ x^o \Rightarrow \vec{u}; \sigma \dashv \Delta \quad o = x \cdot o'}{\Gamma \vdash o(\vec{t}) :> N(x) \Rightarrow o'(\vec{u}); \sigma \dashv \Delta} \\[2ex] \dfrac{o \neq x \cdot \ldots}{\Gamma \vdash o(\vec{t}) :> N(x) \Rightarrow !; ! \dashv \bot} \end{array}\right\} \text{Examples in next slides}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Pruning $o(\vec{t})$: variable case

o is a variable $\Rightarrow \vec{t} = ()$.

$$\frac{\Gamma \vdash \overbrace{\vec{t}}^{()} :> \ldots \quad \overbrace{o = \vec{x} \cdot o'}^{o = x_{o'}}}{\Gamma \vdash o(\vec{t}) :> N(\vec{x}) \Rightarrow o'(\vec{u}); \sigma \dashv \Delta} \quad \Leftrightarrow \quad \frac{o = x_i}{\Gamma \vdash o :> N(\vec{x}) \Rightarrow i; id_\Gamma \dashv \Gamma}$$

$$\frac{o \neq x \cdot \ldots}{\Gamma \vdash o(\vec{t}) :> N(x) \Rightarrow !; ! \dashv \bot} \quad \Leftrightarrow \quad \frac{o \notin \vec{x}}{\Gamma \vdash o :> N(\vec{x}) \Rightarrow !; ! \dashv \bot}$$

A. Lafont, N. Krishnaswami   Generic pattern unification

Pruning $o(\vec{t})$: operation case

Assume $o : (\alpha_1, \ldots, \alpha_p)$

$$\frac{\Gamma \vdash \vec{t} :> \overbrace{\mathcal{L}^+ x^o}^{M_1(\vec{x}, n, \ldots, n+\alpha_1-1), \ldots, M_p(\ldots)} \Rightarrow \vec{u}; \sigma \dashv \Delta \qquad \overbrace{o = x \cdot o'}^{o=o'}}{\Gamma \vdash o(\vec{t}) :> N(x) \Rightarrow o'(\vec{u}); \sigma \dashv \Delta}$$

$$\frac{o \neq x \cdot \ldots}{\Gamma \vdash o(\vec{t}) :> N(x) \Rightarrow !; ! \dashv \bot} \qquad \text{never applies}$$

A. Lafont, N. Krishnaswami   Generic pattern unification