Generic pattern unification

A categorical approach

Ambroise Lafont Neel Krishnaswami

University of Cambridge

October 2022

What is unification?

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

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

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

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

Indecidable

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

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

Outline

Outline

Outline

Syntax (De Bruijn levels)

Metavariable context $(M_1 : n_1, \dots)$

$$\overbrace{\Gamma}\ ;\ \underbrace{n}\ \vdash t$$

Variable context

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

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

Metavariable substitution

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

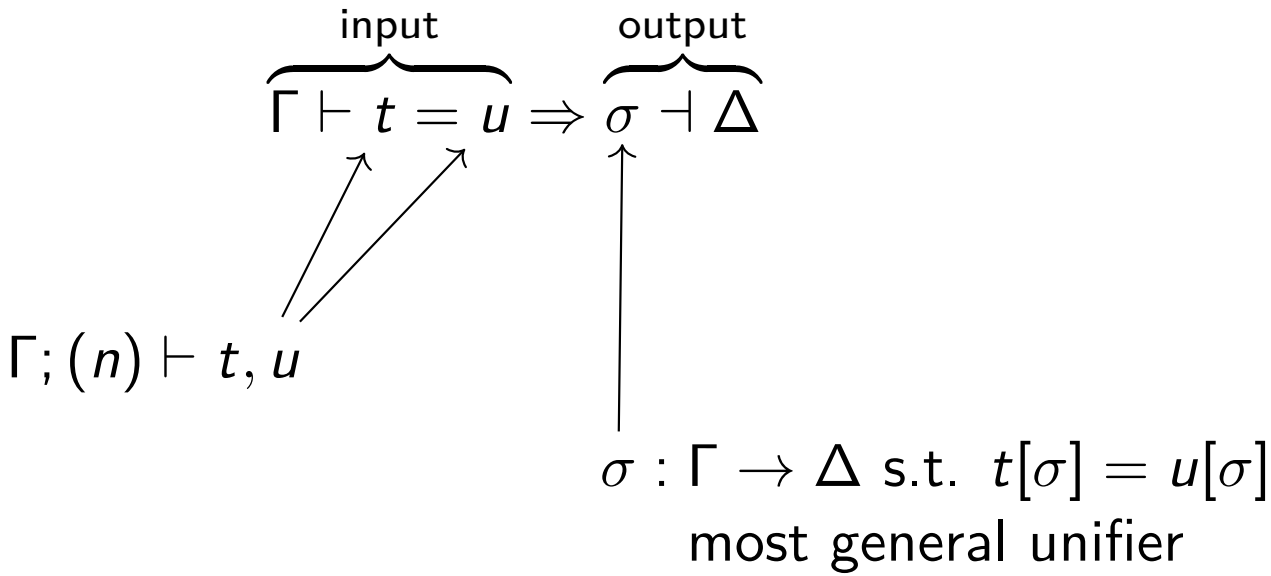
$$(\sigma_1, \ldots, \sigma_p) \qquad \text{s.t.} \qquad \Delta; m_i \vdash \sigma_i$$

σ extends to terms:

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

Outline

Unification algorithm



Examples

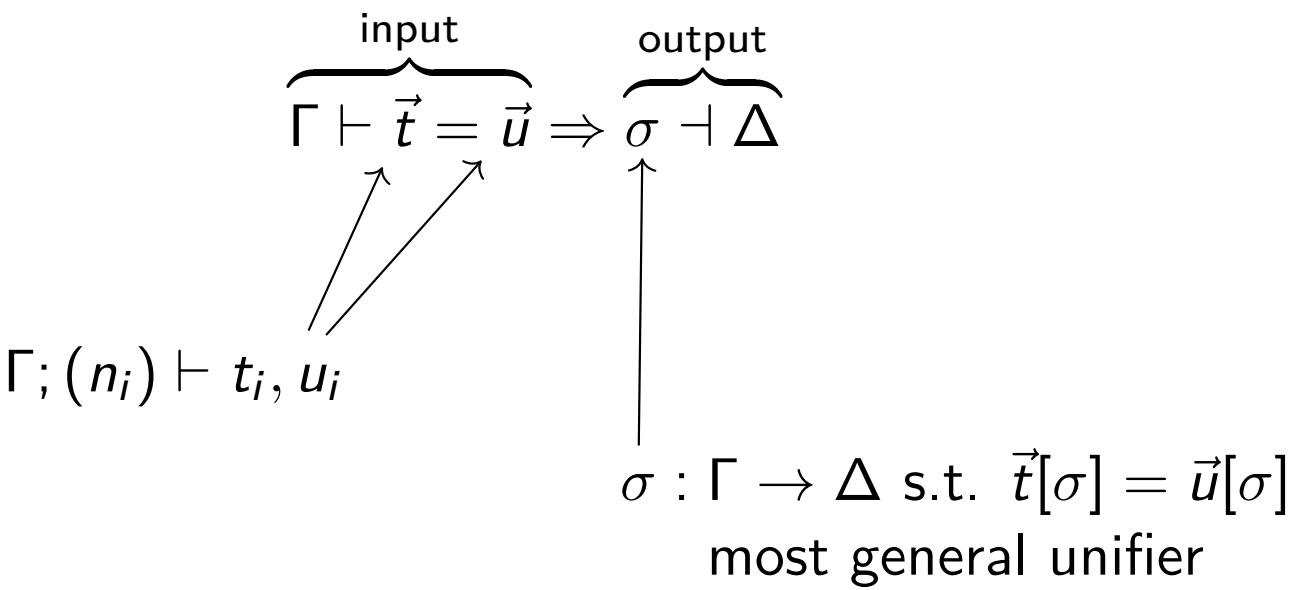
$$\Gamma, M : 2 \vdash M(x, y) = x \Rightarrow (M \mapsto 0) \dashv \Gamma$$

$$\Gamma, M : 2 \vdash M(x, y) = y \Rightarrow (M \mapsto 1) \dashv \Gamma$$

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

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

Unifying lists of terms

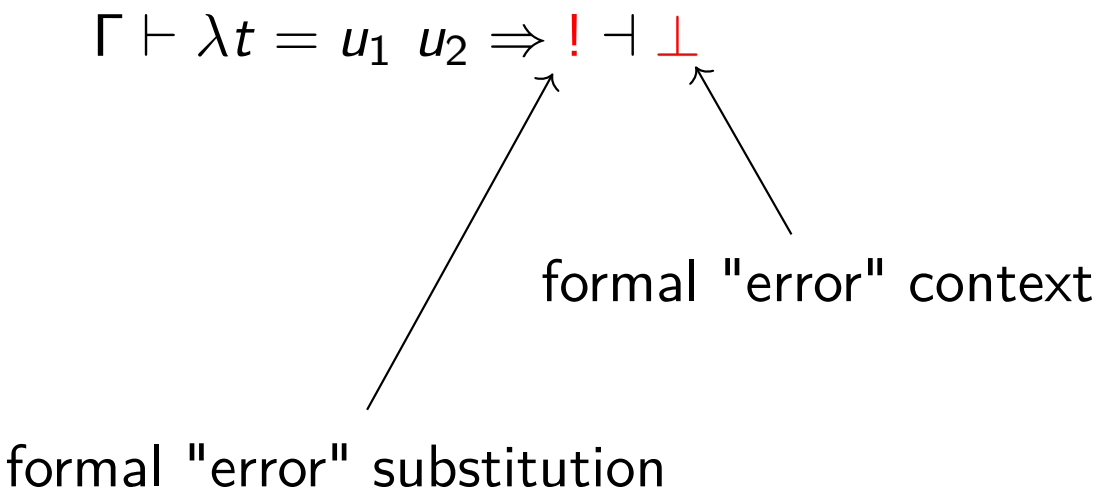


Examples (lists)

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

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

Impossible cases



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 $M \notin u$

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,

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

Cyclic case

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

No unifier (sizes cannot match after substitution)

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

sizes cannot match after substitution

Non cyclic case

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

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

- Requires

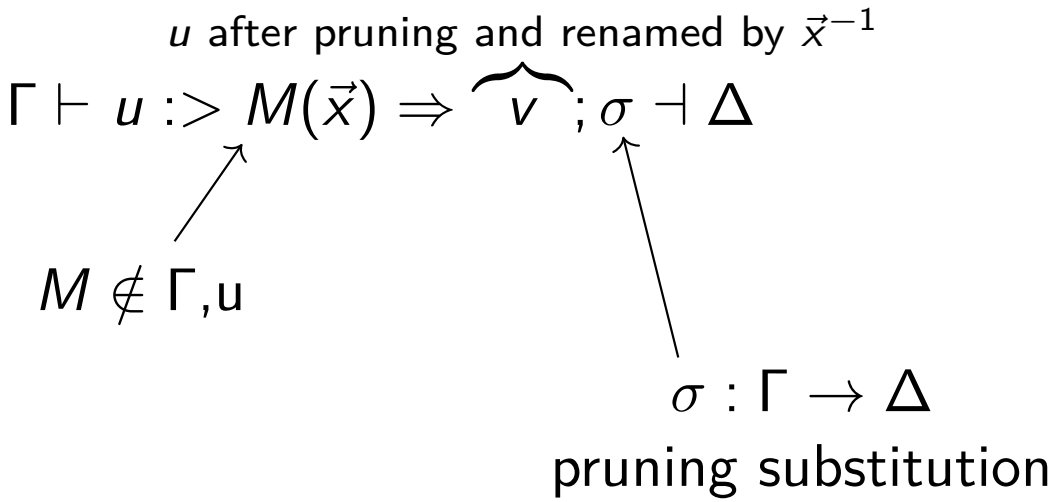
$$fv(u) \subset \vec{x} \tag{1}$$

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

Example

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

Pruning phase



Intuition: $\sigma, M \mapsto v =$ most general unifier for $u \stackrel{?}{=} M(\vec{x})$

'v replaces M'

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

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

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

Pruning: other examples

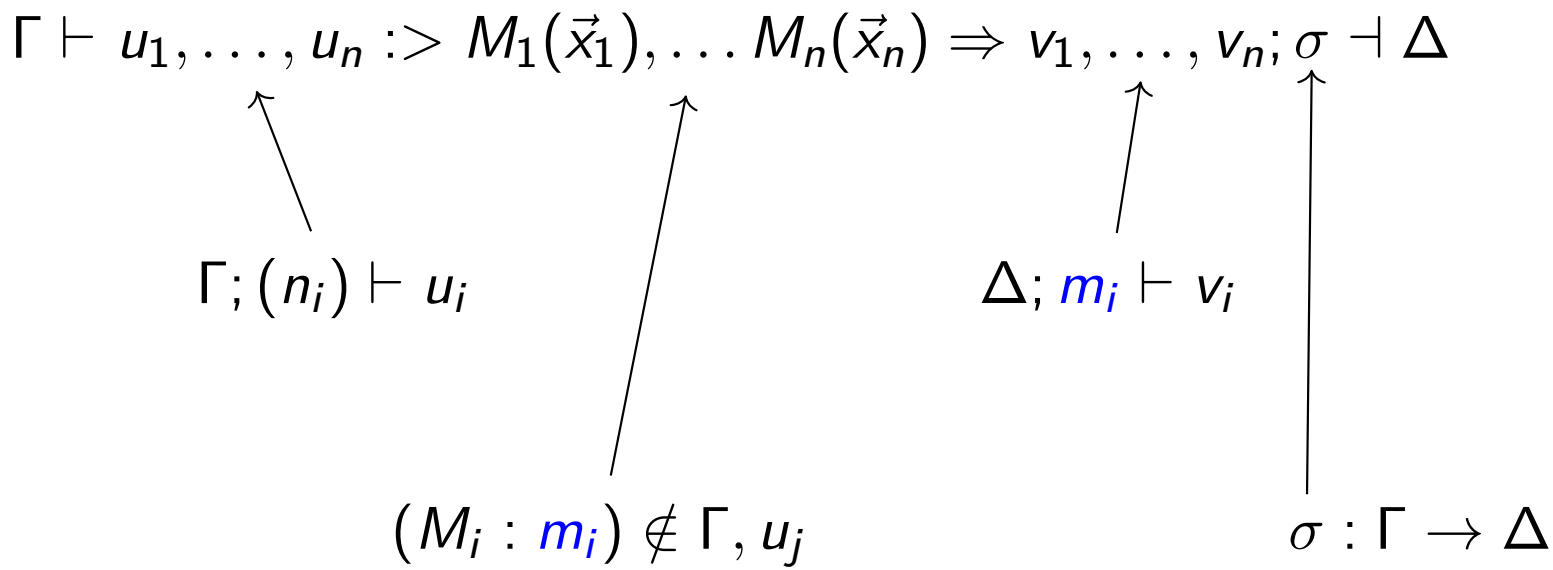
$$\frac{}{\Gamma \vdash x_i :> M(x_0, \ldots, x_n) \Rightarrow i; id_\Gamma \dashv \Gamma} \qquad \frac{y \notin \vec{x}}{\Gamma \vdash y :> M(\vec{x}) \Rightarrow !; ! \dashv \bot}$$

bound variable

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

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

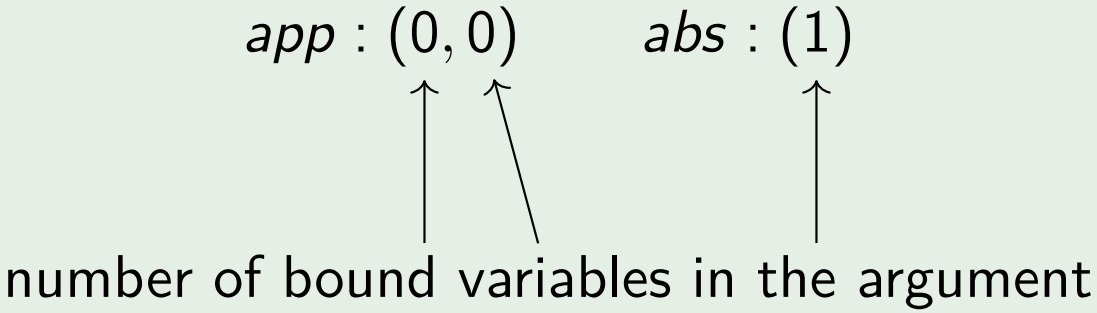
Pruning multi-terms



Outline

Parameterisation by a *signature*

Binding signature for pure λ-calculus



Example: pruning an operation

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

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

Semantics

$$\Gamma \vdash t = u \Rightarrow \sigma \dashv \Delta \quad \Leftrightarrow \quad \sigma \quad = \textbf{coequaliser of } t \text{ and } u$$
$$\Gamma \vdash t :> f \Rightarrow v; \sigma \dashv \Delta \quad \Leftrightarrow \quad \sigma, v \quad = \textbf{pushout of } t \text{ and } f$$

... in the *second-order algebraic theory*[1] of S (cf next section)

[1] [Fiore-Mahmoud 2010]

Outline

Outline

Pure λ-calculus as a functor

$\mathbb{F}_m :=$ 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 $[\mathbb{F}_m, \mathrm{Set}]$

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

Pure λ-calculus with a metavariable

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

As a fixpoint:

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

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

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

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 $yn \overset{t}{\underset{u}{\rightrightarrows}} T\underline{\Gamma}$

in $\mathrm{Th}_F =$ Kleisli category restricted to finite coproducts of representable functors ($\underline{\Gamma}$).

Outline

Signature

1. $\mathcal{A}$ small category (e.g., $\mathbb{F}_m$) s.t.
   - all morphisms are monic (pattern restriction)
   - $\mathcal{A}$ has finite connected limits ($M(\vec{x}) \stackrel{?}{=} N(\vec{y})$)
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.

Semantics of unification

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

Proof: By proving termination of the unification algorithm.

Outline

Example: System F

- Objects of $\mathcal{A}$:

$$n|\underbrace{\tau_1, \ldots, \tau_p \vdash \tau_f}_{\text{Types taking free variables in } \{0, \ldots, n-1\}}$$

$\Rightarrow$ System F defines a functor $\Phi : \mathcal{A} \to \mathrm{Set}$

$$\Phi(n|\tau_1, \ldots, \tau_p \vdash \tau_f) = \{t \text{ s.t. } \mathsf{n}|\tau_1, \ldots, \tau_p \vdash t : \tau_f\}$$

- Endofunctor F on $[\mathcal{A}, \mathrm{Set}]$ s.t. $\Phi = \mu X.F(X)$?
- Metavariables?

The endofunctor for system F

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

Typing rule	$F(X)_{n\|\Gamma \vdash \tau} = \coprod \ldots$
$\dfrac{n+1 \| wk(\Gamma) \vdash t : \tau'}{n \| \Gamma \vdash \Lambda t : \Lambda \tau'}$ T-Abs	$\coprod_{\tau' \text{ s.t. } \tau = \Lambda \tau'} X_{n+1\|wk(\Gamma) \vdash \tau'}$
$\dfrac{n \| \Gamma \vdash t : \Lambda \tau_1}{n \| \Gamma \vdash t \cdot \tau_2 : \tau_1[\tau_2]}$ T-App	$\coprod_{\tau_1, \tau_2 \text{ s.t. } \tau = \tau_1[\tau_2]} X_{n\|\Gamma \vdash \Lambda \tau_1}$
$\dfrac{n \| \Gamma \vdash t : \tau' \Rightarrow \tau \quad n \| \Gamma \vdash u : \tau'}{n \| \Gamma \vdash t\ u : \tau}$ App	$\coprod_{\tau'} X_{n\|\Gamma \vdash \tau' \Rightarrow \tau} \times X_{n\|\Gamma \vdash \tau'}$
$\ldots$	$\ldots$

- System F = initial algebra for F

Metavariables in system F

$$\frac{\alpha_1, \ldots, \alpha_p \text{ distinct}, < n \qquad x_1, \ldots x_q \text{ distinct} \qquad \tau_i[\vec{\alpha}] = \Gamma_i}{\underbrace{\Delta, M : (p|\tau_1, \ldots, \tau_q \vdash \tau_f)}_{\text{Metavariable context}}; n|\Gamma \vdash M(\underbrace{\vec{\alpha}}_{\text{type variables}}, \vec{x}) : \tau_f[\vec{\alpha}]}$$

Unification in system F

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

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.

$$x'_{\vec{z}} = y'_{\vec{z}}$$

where $\vec{x}' = \vec{x}[\vec{\gamma}^{-1}]$ and $\vec{y}' = \vec{y}[\vec{\gamma}^{-1}]$

Outline

Interpreting the unification statements

Notations

$$\Gamma \vdash t = u \Rightarrow \sigma \dashv \Delta \quad \Leftrightarrow \quad \sigma \quad = \textbf{coequaliser of } t \text{ and } u$$
$$\Gamma \vdash t :> f \Rightarrow v; \sigma \dashv \Delta \quad \Leftrightarrow \quad \sigma, v \quad = \textbf{pushout of } t \text{ and } f$$

mostly used in $\mathrm{Th}_F^* = \mathrm{Th}_F +$ a free terminal object $\bot$.

Summary of the generic unification algorithm

$$\frac{}{\Gamma \vdash () = () \Rightarrow 1_\Gamma \dashv \Gamma} \qquad \frac{}{\bot \vdash \vec{t} = \vec{u} \Rightarrow ! \dashv \bot}$$

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

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

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

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

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

Pruning phase

$$\frac{}{\Gamma \vdash () :> () \Rightarrow (); 1_\Gamma \dashv \Gamma} \qquad \frac{}{\bot \vdash \vec{t} :> \vec{f} \Rightarrow !; ! \dashv \bot}$$

$$\frac{\Gamma \vdash t_1 :> f_1 \Rightarrow u_1; \sigma_1 \dashv \Delta_1 \quad \Delta_1 \vdash \vec{t_2}[\sigma_1] :> \vec{f_2} \Rightarrow \vec{u_2}; \sigma_2 \dashv \Delta_2}{\Gamma \vdash t_1, \vec{t_2} :> f_1 + \vec{f_2} \Rightarrow u_1[\sigma_2], \vec{u_2}; \sigma_1[\sigma_2] \dashv \Delta_2}\text{P-Split}$$

$$\frac{\Gamma \vdash \vec{t} :> \mathcal{L}^+ x^o \Rightarrow \vec{u}; \sigma \dashv \Delta \qquad o = x \cdot o'}{\Gamma \vdash o(\vec{t}) :> N(x) \Rightarrow o'(\vec{u}); \sigma \dashv \Delta}\text{P-Rig}$$

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

$$\frac{c \vdash_{\mathscr{D}} y :> x \Rightarrow y'; x' \dashv d}{\Gamma, M : c \vdash M(y) :> N(x) \Rightarrow M'(y'); M \mapsto M'(x') \dashv \Gamma, M' : d}\text{P-Flex}$$