Quiz

Assuming that a and b are distinct variables, is it possible to find λ-terms $M_1..M_7$ that make the following pairs α-equivalent?

- $\lambda a.\lambda b.(M_1\ b)$ and $\lambda b.\lambda a.(a\ M_1)$
- $\lambda a.\lambda b.(M_2\ b)$ and $\lambda b.\lambda a.(a\ M_3)$
- $\lambda a.\lambda b.(b\ M_4)$ and $\lambda b.\lambda a.(a\ M_5)$
- $\lambda a.\lambda b.(b\ M_6)$ and $\lambda a.\lambda a.(a\ M_7)$

If there is one solution for a pair, can you describe all its solutions?

Nominal Techniques Course

Friday-Lecture

Christian Urban

University of Cambridge

Nominal Unification

What? Unific. for schematic-variables and binders

$$\frac{\mathtt{app}(\mathtt{fn}\, \boldsymbol{a}.\boldsymbol{Y}, \boldsymbol{X}) \Downarrow \boldsymbol{V}}{\mathtt{let}\ \boldsymbol{a} = \boldsymbol{X}\ \mathtt{in}\ \boldsymbol{Y} \Downarrow \boldsymbol{V}}$$

Why?

- First-order unification is simple, but cannot be used for terms involving binders.
- Higher-order unification is more complicated, and for schematic-variables has several drawbacks e.g., capture-avoiding substitution . . .

Substitution

Schematic variables work with possibly-capturing substitutions, e.g.

$$\frac{\mathrm{app}(\mathtt{fn}\; \boldsymbol{a}.\boldsymbol{Y}, \boldsymbol{X}) \Downarrow \boldsymbol{V}}{\mathtt{let}\; \boldsymbol{a} = \boldsymbol{X}\; \mathtt{in}\; \boldsymbol{Y} \Downarrow \boldsymbol{V}} \quad \text{scheme}$$

Substitution

Schematic variables work with possibly-capturing substitutions, e.g.

$$\frac{\mathrm{app}(\mathtt{fn}\,\boldsymbol{a}.\boldsymbol{Y}, \boldsymbol{X}) \Downarrow \boldsymbol{V}}{\mathtt{let}\,\boldsymbol{a} = \boldsymbol{X}\,\mathtt{in}\,\boldsymbol{Y} \Downarrow \boldsymbol{V}} \quad \text{scheme}$$

$\mathtt{let}\,\boldsymbol{a} = \mathbf{1}\,\mathtt{in}\,\boldsymbol{a} \Downarrow \mathbf{1}$

Substitution

Schematic variables work with possibly-capturing substitutions, e.g.

$$\frac{\mathrm{app}(\mathrm{fn}\, \boldsymbol{a}.\boldsymbol{Y}, \boldsymbol{X}) \Downarrow \boldsymbol{V}}{\mathtt{let}\ \boldsymbol{a} = \boldsymbol{X}\ \mathtt{in}\ \boldsymbol{Y} \Downarrow \boldsymbol{V}}$$

scheme

$\mathtt{let}\ \boldsymbol{a} = \mathbf{1}\ \mathtt{in}\ \boldsymbol{a} \Downarrow \mathbf{1}$

$[\boldsymbol{Y} := \boldsymbol{a}; \boldsymbol{X}, \boldsymbol{V} := 1]$

correct instance

$$\frac{\mathrm{app}(\mathrm{fn}\, \boldsymbol{a}.\boldsymbol{a}, 1) \Downarrow \mathbf{1}}{\mathtt{let}\ \boldsymbol{a} = \mathbf{1}\ \mathtt{in}\ \boldsymbol{a} \Downarrow \mathbf{1}}$$

Substitution

Schematic variables work with possibly-capturing substitutions, e.g.

$$\frac{\mathrm{app}(\mathrm{fn}\, a.Y, X) \Downarrow V}{\mathtt{let}\; a = X \;\mathtt{in}\; Y \Downarrow V}$$ scheme

$$\mathtt{let}\; a = 1 \;\mathtt{in}\; a \Downarrow 1 \quad =_\alpha \quad \mathtt{let}\; b = 1 \;\mathtt{in}\; b \Downarrow 1$$

$$[Y := a; X, V := 1] \qquad [Y := b; X, V := 1]$$

correct instance

$$\frac{\mathrm{app}(\mathrm{fn}\, a.a, 1) \Downarrow 1}{\mathtt{let}\; a = 1 \;\mathtt{in}\; a \Downarrow 1}$$

Substitution

Schematic variables work with possibly-capturing substitutions, e.g.

$$\frac{\mathrm{app}(\mathrm{fn}\, a.Y, X) \Downarrow V}{\mathtt{let}\ a = X\ \mathtt{in}\ Y \Downarrow V}$$ scheme

$$\mathtt{let}\ a = 1\ \mathtt{in}\ a \Downarrow 1 \quad =_\alpha \quad \mathtt{let}\ b = 1\ \mathtt{in}\ b \Downarrow 1$$

$$[Y := a; X, V := 1] \qquad [Y := b; X, V := 1]$$

correct instance

$$\frac{\mathrm{app}(\mathrm{fn}\, a.a, 1) \Downarrow 1}{\mathtt{let}\ a = 1\ \mathtt{in}\ a \Downarrow 1}$$

incorrect instance

$$\frac{\mathrm{app}(\mathrm{fn}\, a.b, 1) \Downarrow 1}{\mathtt{let}\ b = 1\ \mathtt{in}\ b \Downarrow 1}$$

HOAS

This style of reasoning can be made precise by using higher-order abstract syntax (HOAS) and higher-order unification (capture-avoiding substitutions).

$$\frac{\texttt{app}\;(\texttt{fn}\;\boldsymbol{\lambda a.F(a)})\;\boldsymbol{X} \Downarrow \boldsymbol{V}}{\texttt{let}\;\boldsymbol{X}\;\boldsymbol{\lambda a.F(a)} \Downarrow \boldsymbol{V}}\quad \boldsymbol{\lambda}\text{-calc.}$$

HOAS

This style of reasoning can be made precise by using higher-order abstract syntax (HOAS) and higher-order unification (capture-avoiding substitutions).

$$\frac{\texttt{app}\,(\texttt{fn}\,\boldsymbol{F})\;\boldsymbol{X} \Downarrow \boldsymbol{V}}{\texttt{let}\;\boldsymbol{X}\;\boldsymbol{F} \Downarrow \boldsymbol{V}} \quad \lambda\text{-calc.}$$

HOAS

This style of reasoning can be made precise by using higher-order abstract syntax (HOAS) and higher-order unification (capture-avoiding substitutions).

$$\frac{\mathtt{app}\ (\mathtt{fn}\ \boldsymbol{F})\ \boldsymbol{X} \Downarrow \boldsymbol{V}}{\mathtt{let}\ \boldsymbol{X}\ \boldsymbol{F} \Downarrow \boldsymbol{V}}$$ λ-calc.

$\mathtt{let}\ \mathbf{1}\ \boldsymbol{\lambda a.a} \Downarrow \mathbf{1}$ $\mathtt{let}\ \mathbf{1}\ \boldsymbol{\lambda b.b} \Downarrow \mathbf{1}$

$$\frac{\mathtt{app}(\mathtt{fn}\ \boldsymbol{\lambda a.a})\ \mathbf{1} \Downarrow \mathbf{1}}{\mathtt{let}\ \mathbf{1}\ \boldsymbol{\lambda a.a} \Downarrow \mathbf{1}} \qquad \frac{\mathtt{app}(\mathtt{fn}\ \boldsymbol{\lambda b.b})\ \mathbf{1} \Downarrow \mathbf{1}}{\mathtt{let}\ \mathbf{1}\ \boldsymbol{\lambda b.b} \Downarrow \mathbf{1}}$$

HOAS

Thi y
usi
an
sul

le

a 1

Drawbacks:

- we targeted α, but have to deal with β (or Miller's β_0, at least) as well
- unification theory is not simple
- informal practice suggests that leaving name dependencies implicit can be convenient
- combining HOAS and structural induction can be a nightmare

Do we have to put up with them? No!

Terms

- $\langle\rangle$ Units
- $\langle t, t' \rangle$ Pairs
- $F\,t$ Funct.

Terms

- $\langle\rangle$ Units
- $\langle t, t'\rangle$ Pairs
- $F\,t$ Funct.
- a Atoms
- $a.t$ Abstractions

bindable names (of object-level variables etc.)

generic binder:

$\ulcorner\lambda a.a\urcorner \mapsto \mathtt{fn}\ a.a$

constructions like $\mathtt{fn}\ X.X$ are not allowed

Terms

- $\langle\rangle$ Units
- $\langle t, t'\rangle$ Pairs
- $F\,t$ Funct.
- a Atoms
- $a.t$ Abstractions
- $\pi{\cdot}X$ Suspensions

Terms

- $\langle\rangle$ Units
- a Atoms
- $\langle t, t'\rangle$ Pairs
- $a.t$ Abstractions
- $F\,t$ Funct.
- $\pi{\cdot}X$ Suspensions

π is an explicit permutation, which is a list of swappings $(a_1\,b_1)\ldots(a_n\,b_n)$, waiting to be applied to the term that is substituted for X

X is a variable standing for an unknown term

Permutations

a permutation applied to a term:

- $$[]\bullet a \stackrel{\text{def}}{=} a$$
- $$(b\,c)::\pi \bullet a \stackrel{\text{def}}{=} \begin{cases} c & \text{if } \pi\bullet a = b \\ b & \text{if } \pi\bullet a = c \\ \pi\bullet a & \text{otherwise} \end{cases}$$

Permutations

a permutation applied to a term:

- $$[]\bullet a \stackrel{\text{def}}{=} a$$
- $$(b\,c)::\pi\bullet a \stackrel{\text{def}}{=} \begin{cases} c & \text{if } \pi\bullet a = b \\ b & \text{if } \pi\bullet a = c \\ \pi\bullet a & \text{otherwise} \end{cases}$$
- $$\pi\bullet a.t \stackrel{\text{def}}{=} \pi\bullet a.\pi\bullet t$$

Permutations

a permutation applied to a term:

- $$[]\bullet a \stackrel{\text{def}}{=} a$$
- $$(b\,c)::\pi\bullet a \stackrel{\text{def}}{=} \begin{cases} c & \text{if } \pi\bullet a = b \\ b & \text{if } \pi\bullet a = c \\ \pi\bullet a & \text{otherwise} \end{cases}$$
- $$\pi\bullet a.t \stackrel{\text{def}}{=} \pi\bullet a.\pi\bullet t$$
- $$\pi\bullet\pi'\cdot X \stackrel{\text{def}}{=} (\pi @ \pi')\cdot X$$

Freshness Relation

We will identify

$$\mathtt{fn}\, a.X \;\approx\; \mathtt{fn}\, b.(a\, b)\cdot X$$

provided that 'b is fresh for X — ($b \# X$)', i.e., does not occur freely in any ground term that might be substituted for X.

Freshness Relation

We will identify

$$\mathtt{fn}\, a.X \;\approx\; \mathtt{fn}\, b.(a\, b)\cdot X$$

provided that 'b is fresh for X — $(b \# X)$',
i.e., does not occur
that might be subs

explicit permutation — waits to be applied to the term that is substituted for X

Freshness Relation

We will identify

$$\mathtt{fn}\, a.X \;\approx\; \mathtt{fn}\, b.(a\,b)\cdot X$$

provided that 'b is fresh for X — ($b \# X$)',
i.e., does not occur freely in any ground term
that might be substituted for X.

Freshness Relation

We will identify

$$\mathtt{fn}\, a.X \;\approx\; \mathtt{fn}\, b.(a\, b)\cdot X$$

provided that 'b is fresh for X — ($b \;\#\; X$)', i.e., does not occur freely in any ground term that might be substituted for X.

If we know more about X, e.g., if we knew that $a \;\#\; X$ and $b \;\#\; X$, then we can replace $(a\, b)\cdot X$ by X.

Freshness Assumptions

Our equality is **not** just

$$t \approx t' \qquad \alpha\text{-equivalence}$$

Freshness Assumptions

but judgements

$$\nabla \vdash t \approx t'$$ α-equivalence

where

$$\nabla = \{a_1 \# X_1, \ldots, a_n \# X_n\}$$

is a finite set of freshness assumptions.

Freshness Assumptions

but judgements

$$\nabla \vdash t \approx t'$$ α-equivalence

where

$$\nabla = \{a_1 \# X_1, \ldots, a_n \# X_n\}$$

is a finite set of freshness assumptions.

$$\{a \# X, b \# X\} \vdash \mathtt{fn}\, a.X \approx \mathtt{fn}\, b.X$$

Freshness Assumptions

but judgements

$\nabla \vdash t \approx t'$ α-equivalence

$\nabla \vdash a \# t$ freshness

where

$$\nabla = \{a_1 \# X_1, \ldots, a_n \# X_n\}$$

is a finite set of freshness assumptions.

$$\begin{aligned} \{b \# X\} &\vdash b \# a.X \\ \{\} &\vdash a \# a.X \end{aligned}$$

Rules for Equivalence

Excerpt
(i.e. only the interesting rules)

Rules for Equivalence

$$\frac{\nabla \vdash t \approx t'}{\nabla \vdash a.t \approx a.t'}$$

$$\frac{a \neq b \qquad \nabla \vdash t \approx (a\,b)\bullet t' \qquad \nabla \vdash a \# t'}{\nabla \vdash a.t \approx b.t'}$$

Rules for Equivalence

$$\frac{(a \;\#\; X) \in \nabla \quad \text{for all } a \text{ with } \pi \bullet a \neq \pi' \bullet a}{\nabla \vdash \pi{\cdot}X \approx \pi'{\cdot}X}$$

Rules for Equivalence

$$\frac{(a \# X) \in \nabla \quad \text{for all } a \text{ with } \pi \bullet a \neq \pi' \bullet a}{\nabla \vdash \pi \cdot X \approx \pi' \cdot X}$$

for example

$$\{a \# X, b \# X\} \vdash X \approx (a\, b) \cdot X$$

Rules for Equivalence

$$\frac{(a \# X) \in \nabla \quad \text{for all } a \text{ with } \pi \bullet a \neq \pi' \bullet a}{\nabla \vdash \pi \cdot X \approx \pi' \cdot X}$$

for example

$$\{a \# X, c \# X\} \vdash (a\, c)(a\, b) \cdot X \approx (b\, c) \cdot X$$

because $(a\, c)(a\, b)$: $a \mapsto b$, $b \mapsto c$, $c \mapsto a$ $\quad$ $(b\, c)$: $a \mapsto a$, $b \mapsto c$, $c \mapsto b$

disagree at a and c.

Rules for Freshness

Excerpt
(again only the interesting rules)

Rules for Freshness

$$\frac{a \neq b}{\nabla \vdash a \,\#\, b}$$

$$\frac{}{\nabla \vdash a \,\#\, a.t} \qquad \frac{a \neq b \qquad \nabla \vdash a \,\#\, t}{\nabla \vdash a \,\#\, b.t}$$

$$\frac{(\pi^{-1}{\bullet}a \,\#\, X) \in \nabla}{\nabla \vdash a \,\#\, \pi{\cdot}X}$$

$\approx$ is an Equivalence

Theorem: $\approx$ is an equivalence relation.

(Reflexivity)	$\nabla \vdash t \approx t$
(Symmetry)	if $\nabla \vdash t_1 \approx t_2$ then $\nabla \vdash t_2 \approx t_1$
(Transitivity)	if $\nabla \vdash t_1 \approx t_2$ and $\nabla \vdash t_2 \approx t_3$ then $\nabla \vdash t_1 \approx t_3$

$\approx$ is an Equivalence

Theorem: $\approx$ is an equivalence relation.

because $\approx$ has very good properties:

- $\nabla \vdash t \approx t'$ then $\nabla \vdash \pi \bullet t \approx \pi \bullet t'$
- $\nabla \vdash a \# t$ then $\nabla \vdash \pi \bullet a \# \pi \bullet t$

$\approx$ is an Equivalence

Theorem: $\approx$ is an equivalence relation.

because $\approx$ has very good properties:

- $\nabla \vdash t \approx t'$ then $\nabla \vdash \pi \bullet t \approx \pi \bullet t'$
- $\nabla \vdash a \# t$ then $\nabla \vdash \pi \bullet a \# \pi \bullet t$
- $\nabla \vdash t \approx \pi \bullet t'$ then $\nabla \vdash (\pi^{-1}) \bullet t \approx t'$
- $\nabla \vdash a \# \pi \bullet t$ then $\nabla \vdash (\pi^{-1}) \bullet a \# t$

$\approx$ is an Equivalence

Theorem: $\approx$ is an equivalence relation.

because $\approx$ has very good properties:

- $\nabla \vdash t \approx t'$ then $\nabla \vdash \pi{\bullet}t \approx \pi{\bullet}t'$
- $\nabla \vdash a \# t$ then $\nabla \vdash \pi{\bullet}a \# \pi{\bullet}t$
- $\nabla \vdash t \approx \pi{\bullet}t'$ then $\nabla \vdash (\pi^{-1}){\bullet}t \approx t'$
- $\nabla \vdash a \# \pi{\bullet}t$ then $\nabla \vdash (\pi^{-1}){\bullet}a \# t$
- $\nabla \vdash a \# t$ and $\nabla \vdash t \approx t'$ then $\nabla \vdash a \# t'$

Comparison with $=_\alpha$

Traditionally $=_\alpha$ is defined as

> least congruence which identifies $a.t$ with $b.[a := b]t$ provided b is not free in t

where $[a := b]t$ replaces all free occurrences of a by b in t.

Comparison with $=_\alpha$

Traditionally $=_\alpha$ is defined as

> least congruence which identifies $a.t$ with $b.[a := b]t$ provided b is not free in t

where $[a := b]t$ replaces all free occurrences of a by b in t.

For ground terms:

> Theorem: $t =_\alpha t'$ iff $\varnothing \vdash t \approx t'$
>
> $a \not\in FA(t)$ iff $\varnothing \vdash a \# t$

Comparison with $=_\alpha$

Traditionally $=_\alpha$ is defined as

least congruence which identifies $a.t$ with $b.[a := b]t$ provided b is not free in t

where $[a := b]t$ replaces all free occurrences of a by b in t.

In general $=_\alpha$ and $\approx$ are distinct!

$a.X =_\alpha b.X$ but not

$\varnothing \vdash a.X \approx b.X$ ($a \neq b$)

Comparison with $=_\alpha$

That is a crucial point: if we had

$$\varnothing \vdash a.X \approx b.X\,,$$

then applying $[X := a]$, $[X := b]$, ...
give two terms that are **not** α-equivalent.

The freshness constraints $a \# X$ and $b \# X$ rule out the problematic substitutions. Therefore

$$\{a \# X, b \# X\} \vdash a.X \approx b.X$$

does hold.

Substitutions

- $\sigma(a.t) \stackrel{\text{def}}{=} a.\sigma(t)$
- $\sigma(\pi{\cdot}X) \stackrel{\text{def}}{=} \begin{cases} \pi \bullet \sigma(X) & \text{if } \sigma(X) \neq X \\ \pi{\cdot}X & \text{o'wise do nothing} \end{cases}$

Substitutions

- $\sigma(a.t) \stackrel{\text{def}}{=} a.\sigma(t)$
- $\sigma(\pi{\cdot}X) \stackrel{\text{def}}{=} \begin{cases} \pi \bullet \sigma(X) & \text{if } \sigma(X) \neq X \\ \pi{\cdot}X & \text{o'wise do nothing} \end{cases}$

for example

$$a.(a\,b){\cdot}X \;\; [X := \langle b, Y\rangle]$$

Substitutions

- $\sigma(a.t) \stackrel{\text{def}}{=} a.\sigma(t)$
- $\sigma(\pi{\cdot}X) \stackrel{\text{def}}{=} \begin{cases} \pi \bullet \sigma(X) & \text{if } \sigma(X) \neq X \\ \pi{\cdot}X & \text{o'wise do nothing} \end{cases}$

for example

$$\underline{a.(a\,b){\cdot}X \;\; [X := \langle b, Y\rangle]}$$

$$\Rightarrow \quad a.\underline{(a\,b){\cdot}X[X := \langle b, Y\rangle]}$$

Substitutions

- $\sigma(a.t) \stackrel{\text{def}}{=} a.\sigma(t)$
- $\sigma(\pi{\cdot}X) \stackrel{\text{def}}{=} \begin{cases} \pi \bullet \sigma(X) & \text{if } \sigma(X) \neq X \\ \pi{\cdot}X & \text{o'wise do nothing} \end{cases}$

for example

$$a.(a\,b){\cdot}X\ \ [X := \langle b, Y\rangle]$$

$$\Rightarrow \quad a.\underline{(a\,b){\cdot}X[X := \langle b, Y\rangle]}$$

$$\Rightarrow \quad a.(a\,b)\bullet\underline{\langle b, Y\rangle}$$

Substitutions

- $\sigma(a.t) \stackrel{\text{def}}{=} a.\sigma(t)$
- $\sigma(\pi{\cdot}X) \stackrel{\text{def}}{=} \begin{cases} \pi \bullet \sigma(X) & \text{if } \sigma(X) \neq X \\ \pi{\cdot}X & \text{o'wise do nothing} \end{cases}$

for example

$$a.(a\,b){\cdot}X \;\; [X := \langle b, Y\rangle]$$

$$\Rightarrow \quad a.(a\,b){\cdot}X[X := \langle b, Y\rangle]$$

$$\Rightarrow \quad a.\underline{(a\,b)}\bullet\langle b, Y\rangle$$

$$\Rightarrow \quad a.\langle a, (a\,b){\cdot}Y\rangle$$

Substitutions

- $\sigma(a.t) \stackrel{\text{def}}{=} a.\sigma(t)$
- $\sigma(\pi{\cdot}X) \stackrel{\text{def}}{=} \begin{cases} \pi \bullet \sigma(X) & \text{if } \sigma(X) \neq X \\ \pi{\cdot}X & \text{o'wise do nothing} \end{cases}$
- if $\nabla \vdash t \approx t'$ and $\nabla' \vdash \sigma(\nabla)$
  then $\nabla' \vdash \sigma(t) \approx \sigma(t')$

Substitutions

- $\sigma(a.t) \stackrel{\text{def}}{=} a.\sigma(t)$
- $\sigma(\pi{\cdot}X) \stackrel{\text{def}}{=} \begin{cases} \pi \bullet \sigma(X) & \text{if } \sigma(X) \neq X \\ \pi{\cdot}X & \text{o'wise do nothing} \end{cases}$
- if $\nabla \vdash t \approx t'$ and $\nabla' \vdash \sigma(\nabla)$
  then $\nabla' \vdash \sigma(t) \approx \sigma(t')$

this means
$\nabla' \vdash a \# \sigma(X)$
holds for all
$(a \# X) \in \nabla$

Substitutions

- $\sigma(a.t) \stackrel{\text{def}}{=} a.\sigma(t)$
- $\sigma(\pi{\cdot}X) \stackrel{\text{def}}{=} \begin{cases} \pi \bullet \sigma(X) & \text{if } \sigma(X) \neq X \\ \pi{\cdot}X & \text{o'wise do nothing} \end{cases}$
- if $\nabla \vdash t \approx t'$ and $\nabla' \vdash \sigma(\nabla)$

  then $\nabla' \vdash \sigma(t) \approx \sigma(t')$

Substitutions

- $\sigma(a.t) \stackrel{\text{def}}{=} a.\sigma(t)$
- $\sigma(\pi{\cdot}X) \stackrel{\text{def}}{=} \begin{cases} \pi \bullet \sigma(X) & \text{if } \sigma(X) \neq X \\ \pi{\cdot}X & \text{o'wise do nothing} \end{cases}$
- if $\nabla \vdash t \approx t'$ and $\nabla' \vdash \sigma(\nabla)$
  then $\nabla' \vdash \sigma(t) \approx \sigma(t')$
- $\sigma(\pi{\bullet}t) = \pi{\bullet}\sigma(t)$

Equational Problems

An equational problem

$$t \approx ? \, t'$$

is **solved** by

- a substitution σ (terms for variables)
- and a set of freshness assumptions ∇

so that $\nabla \vdash \sigma(t) \approx \sigma(t')$.

Unifying equations may entail solving freshness problems.

E.g. assuming that $a \neq a'$, then

$$a.t \approx ?\ a'.t'$$

can only be solved if

$$t \approx ?\ (a\,a')\bullet t' \quad \text{and} \quad a \#?\ t'$$

can be solved.

Freshness Problems

A freshness problem

$$a \;\#?\; t$$

is solved by

- a substitution σ
- and a set of freshness assumptions ∇

so that $\nabla \vdash a \;\#\; \sigma(t)$.

Reductions

A set of $(t \approx ?\ t')$ and $(a\ \# ?\ t)$ problems can be reduced by

$$\stackrel{\sigma}{\Longrightarrow} \quad \text{or} \quad \stackrel{\nabla}{\Longrightarrow}$$

Reductions

- $\{a.t \approx ?\ b.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$ if $a \neq b$
  $\{t \approx ?\ (a\,b) \bullet t', a \ \#?\ t'\} \cup P$

Reductions

- $\{a.t \approx ?\ b.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$ if $a \neq b$

  $\{t \approx ?\ (a\,b)\bullet t', a \ \#?\ t'\} \cup P$
- $\{a.t \approx ?\ a.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow} \{t \approx ?\ t'\} \cup P$

Reductions

- $\{a.t \approx ?\ b.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$ if $a \neq b$
  $\{t \approx ?\ (a\, b) \bullet t', a \,\#?\ t'\} \cup P$
- $\{a.t \approx ?\ a.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow} \{t \approx ?\ t'\} \cup P$
- $\{\pi \bullet X \approx ?\ \pi' \bullet X\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$
  $\{a \,\#?\ X | a \in ds(\pi, \pi')\} \cup P$

Reductions

- $\{a.t \approx ?\ b.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$ if $a \neq b$
  $\{t \approx ?\ (a\,b)\bullet t', a \ \#?\ t'\} \cup P$
- $\{a.t \approx ?\ a.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow} \{t \approx ?\ t'\} \cup P$
- $\{\pi\bullet X \approx ?\ \pi'\bullet X\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$
  $\{a \ \#?\ X | a \in ds(\pi, \pi')\} \cup P$
- $\{\pi\bullet X \approx ?\ t\} \uplus P \stackrel{\sigma}{\Longrightarrow} \sigma P$
  if X does not occur in t

Reductions

- $\{a.t \approx ?\ b.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$ if $a \neq b$
  $\{t \approx ?\ (a\,b)\bullet t', a \ \#?\ t'\} \cup P$
- $\{a.t \approx ?\ a.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow} \{t \approx ?\ t'\} \cup P$
- $\{\pi \bullet X \approx ?\ \pi' \bullet X\}$ $\sigma = [X := \pi^{-1} \bullet t]$
  $\{a \ \#?\ X \mid a \in ds(\pi, \pi')\} \cup P$
- $\{\pi \bullet X \approx ?\ t\} \uplus P \stackrel{\sigma}{\Longrightarrow} \sigma P$
  if X does not occur in t

Reductions

- $\{a.t \approx? b.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$ if $a \neq b$
  $\{t \approx? (a\,b)\bullet t', a \#? t'\} \cup P$
- $\{a.t \approx? a.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow} \{t \approx? t'\} \cup P$
- $\{\pi\bullet X \approx? \pi'\bullet X\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$
  $\{a \#? X | a \in ds(\pi, \pi')\} \cup P$
- $\{\pi\bullet X \approx? t\} \uplus P \stackrel{\sigma}{\Longrightarrow} \sigma P$
  if X does not occur in t
- $\{a \#? b.t\} \uplus P \stackrel{\varnothing}{\Longrightarrow} \{a \#? t\} \cup P$ if $a \neq b$

Reductions

- $\{a.t \approx ?\; b.t'\} \uplus P \overset{\varepsilon}{\Longrightarrow}$ if $a \neq b$
  $\{t \approx ?\; (a\,b)\bullet t', a \;\#?\; t'\} \cup P$
- $\{a.t \approx ?\; a.t'\} \uplus P \overset{\varepsilon}{\Longrightarrow} \{t \approx ?\; t'\} \cup P$
- $\{\pi\bullet X \approx ?\; \pi'\bullet X\} \uplus P \overset{\varepsilon}{\Longrightarrow}$
  $\{a \;\#?\; X | a \in ds(\pi, \pi')\} \cup P$
- $\{\pi\bullet X \approx ?\; t\} \uplus P \overset{\sigma}{\Longrightarrow} \sigma P$
  if X does not occur in t
- $\{a \;\#?\; b.t\} \uplus P \overset{\varnothing}{\Longrightarrow} \{a \;\#?\; t\} \cup P$ if $a \neq b$
- $\{a \;\#?\; \pi\bullet X\} \uplus P \overset{\nabla}{\Longrightarrow} P$

Reductions

- $\{a.t \approx ?\ b.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$ if $a \neq b$
  $\{t \approx ?\ (a\,b)\bullet t', a \# ?\ t'\} \cup P$
- $\{a.t \approx ?\ a.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow} \{t \approx ?\ t'\} \cup P$
- $\{\pi \bullet X \approx ?\ \pi' \bullet X\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$
  $\{a \# ?\ X | a \in ds(\pi, \pi')\} \cup P$
- $\{\pi \bullet X \approx ?\ t\} \uplus P \stackrel{\sigma}{\Longrightarrow} P$
  if X does not occur in t
- $\{a \# ?\ b.t\} \uplus P \stackrel{\varnothing}{\Longrightarrow} \{a \# ?\ t\} \cup P$ if $a \neq b$
- $\{a \# ?\ \pi \bullet X\} \uplus P \stackrel{\nabla}{\Longrightarrow} P$

$\nabla = \{\pi^{-1} \bullet a \# X\}$

Reductions

A set of $(t \approx ?\ t')$ and $(a \ \#?\ t)$ problems can be reduced by

$$\overset{\sigma}{\Longrightarrow} \quad \text{or} \quad \overset{\nabla}{\Longrightarrow}$$

Reductions

A set of $(t \approx ? \; t')$ and $(a \; \# ? \; t)$ problems can be reduced by

$$\stackrel{\sigma}{\Longrightarrow} \quad \text{or} \quad \stackrel{\nabla}{\Longrightarrow}$$

If there is a reduction

$$P \stackrel{\sigma_1}{\Longrightarrow} \ldots \stackrel{\sigma_n}{\Longrightarrow} P' \stackrel{\nabla_1}{\Longrightarrow} \ldots \stackrel{\nabla_m}{\Longrightarrow} \varnothing$$

then

$$(\sigma_n \circ \ldots \circ \sigma_1, \nabla_1 \cup \ldots \cup \nabla_m)$$

is a most general unifier for P.

Most General Unifiers

Definition: for a unification problem P, a solution (σ_1, ∇_1) is more general than another solution (σ_2, ∇_2), iff there exists a substitution σ with

- $\nabla_2 \vdash \sigma(\nabla_1)$
- $\nabla_2 \vdash \sigma_2 \approx \sigma \circ \sigma_1$

Most General Unifiers

Definition: for a unification problem P, a solution (σ_1, ∇_1) is more general than another solution (σ_2, ∇_2), iff there exists a substitution σ with

- $\nabla_2 \vdash \sigma(\nabla_1)$
- $\nabla_2 \vdash \sigma_2 \approx \sigma \circ \sigma_1$

$\nabla_2 \vdash a \# \sigma(X)$ holds for all $(a \# X) \in \nabla_1$

Most General Unifiers

Definition: for a unification problem P, a solution (σ_1, ∇_1) is more general than another solution (σ_2, ∇_2), iff there exists a substitution σ wi

$\nabla_2 \vdash \sigma_2(X) \approx \sigma(\sigma_1(X))$
holds for all
$X \in \mathrm{dom}(\sigma_2) \cup \mathrm{dom}(\sigma \circ \sigma_1)$

- $\nabla_2 \vdash \sigma(\nabla$
- $\nabla_2 \vdash \sigma_2 \approx \sigma \circ \sigma_1$

Existence of MGUs

<u>Theorem</u>: there is an algorithm which, given a nominal unification problem P, decides whether or not it has a solution (σ, ∇), and returns a most general one if it does.

Existence of MGUs

Theorem: there is an algorithm which, given a nominal unification problem P, decides whether or not it has a solution (σ, ∇), and returns a most general one if it does.

Proof: one can reduce all the equations to 'solved form' first (creating a substitution), and then solve the freshness problems (easy).

Remember the Quiz?

Assuming that a and b are distinct variables, is it possible to find λ-terms M_1 to M_7 that make the following pairs α-equivalent?

- $\lambda a.\lambda b.(M_1\ b)$ and $\lambda b.\lambda a.(a\ M_1)$
- $\lambda a.\lambda b.(M_2\ b)$ and $\lambda b.\lambda a.(a\ M_3)$
- $\lambda a.\lambda b.(b\ M_4)$ and $\lambda b.\lambda a.(a\ M_5)$
- $\lambda a.\lambda b.(b\ M_6)$ and $\lambda a.\lambda a.(a\ M_7)$

If there is one solution for a pair, can you describe all its solutions?

Answers to the Quiz

$\lambda a.\lambda b.(M_1\, b)$ and $\lambda b.\lambda a.(a\, M_1)$

Answers to the Quiz

$$a.b.\langle M_1, b\rangle \;\; \approx? \;\; b.a.\langle a, M_1\rangle$$

Answers to the Quiz

$$a.b.\langle M_1, b\rangle \;\approx ?\; b.a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle M_1, b\rangle \approx ? \; (a\,b)\bullet a.\langle a, M_1\rangle \;,\; a \;\#?\; a.\langle a, M_1\rangle$$

Answers to the Quiz

$$a.b.\langle M_1, b\rangle \;\approx?\; b.a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle M_1, b\rangle \approx? \, b.\langle b, (a\,b)\cdot M_1\rangle \;,\; a \;\#?\; a.\langle a, M_1\rangle$$

Answers to the Quiz

$$a.b.\langle M_1, b\rangle \;\approx?\; b.a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle M_1, b\rangle \approx? \; b.\langle b, (a\, b)\cdot M_1\rangle \;,\; a \,\#?\; a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle M_1, b\rangle \approx? \; \langle b, (a\, b)\cdot M_1\rangle \;,\; a \,\#?\; a.\langle a, M_1\rangle$$

Answers to the Quiz

$$a.b.\langle M_1, b\rangle \;\approx?\; b.a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle M_1, b\rangle \approx? b.\langle b, (a\,b)\cdot M_1\rangle \;,\; a \,\#?\, a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle M_1, b\rangle \approx? \langle b, (a\,b)\cdot M_1\rangle \;,\; a \,\#?\, a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} M_1 \approx? b \;,\; b \approx? (a\,b)\cdot M_1 \;,\; a \,\#?\, a.\langle a, M_1\rangle$$

Answers to the Quiz

$$a.b.\langle M_1, b\rangle \;\approx?\; b.a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle M_1, b\rangle \approx? b.\langle b, (a\,b)\cdot M_1\rangle \;,\; a \,\#?\, a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle M_1, b\rangle \approx? \langle b, (a\,b)\cdot M_1\rangle \;,\; a \,\#?\, a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} M_1 \approx? b \;,\; b \approx? (a\,b)\cdot M_1 \;,\; a \,\#?\, a.\langle a, M_1\rangle$$

$$\stackrel{[M_1:=b]}{\Longrightarrow} b \approx? (a\,b)\bullet b \;,\; a \,\#?\, a.\langle a, b\rangle$$

Answers to the Quiz

$$a.b.\langle M_1, b\rangle \;\approx?\; b.a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle M_1, b\rangle \approx? b.\langle b, (a\,b)\cdot M_1\rangle \;,\; a \,\#?\, a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle M_1, b\rangle \approx? \langle b, (a\,b)\cdot M_1\rangle \;,\; a \,\#?\, a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} M_1 \approx? b \;,\; b \approx? (a\,b)\cdot M_1 \;,\; a \,\#?\, a.\langle a, M_1\rangle$$

$$\stackrel{[M_1:=b]}{\Longrightarrow} b \approx? a \;,\; a \,\#?\, a.\langle a, b\rangle$$

Answers to the Quiz

$$a.b.\langle M_1, b\rangle \;\approx?\; b.a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle M_1, b\rangle \approx? \, b.\langle b, (a\,b)\cdot M_1\rangle \;,\; a \;\#?\; a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle M_1, b\rangle \approx? \, \langle b, (a\,b)\cdot M_1\rangle \;,\; a \;\#?\; a.\langle a, M_1\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} M_1 \approx? \, b \;,\; b \approx? (a\,b)\cdot M_1 \;,\; a \;\#?\; a.\langle a, M_1\rangle$$

$$\stackrel{[M_1:=b]}{\Longrightarrow} b \approx? \, a \;,\; a \;\#?\; a.\langle a, b\rangle$$

$$\Longrightarrow \mathit{FAIL}$$

Answers to the Quiz

$$a.b.\langle M_1, b\rangle \;\approx?\; b.a.\langle a, M_1\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} b.\langle M_1, b\rangle \approx? \, b.\langle b, (a\,b)\cdot M_1\rangle \;,\; a \;\#?\; a.\langle a, M_1\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} \langle M_1, b\rangle \approx? \langle b, (a\,b)\cdot M_1\rangle \;,\; a \;\#?\; a.\langle a, M_1\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} M_1 \approx? b \;,\; b \approx? (a\,b)\cdot M_1 \;,\; a \;\#?\; a.\langle a, M_1\rangle$$

$$\overset{[M_1:=b]}{\Longrightarrow} b \approx? a \;,\; a \;\#?\; a.\langle a, b\rangle$$

$$\Longrightarrow \mathit{FAIL}$$

$\lambda a.\lambda b.(M_1\, b) =_\alpha \lambda b.\lambda a.(a\, M_1)$ has no solution

Answers to the Quiz

$\lambda a.\lambda b.(b\, M_6)$ and $\lambda a.\lambda a.(a\, M_7)$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \;\approx?\; a.a.\langle a, M_7\rangle$$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \approx? \; a.a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle b, M_6\rangle \approx? \; a.\langle a, M_7\rangle$$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \;\approx?\; a.a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle b, M_6\rangle \approx? \; a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle b, M_6\rangle \approx? \; \langle b, (b\,a)\cdot M_7\rangle \;,\; b \;\#?\; \langle a, M_7\rangle$$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \;\approx?\; a.a.\langle a, M_7\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} b.\langle b, M_6\rangle \approx? a.\langle a, M_7\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} \langle b, M_6\rangle \approx? \langle b, (b\,a)\cdot M_7\rangle\;,\; b \;\#?\; \langle a, M_7\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} b \approx? b\;,\; M_6 \approx? (b\,a)\cdot M_7\;,\; b \;\#?\; \langle a, M_7\rangle$$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \ \approx? \ a.a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle b, M_6\rangle \approx? \ a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle b, M_6\rangle \approx? \ \langle b, (b\,a)\cdot M_7\rangle \ , \ b \ \#? \ \langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b \approx? \ b \ , \ M_6 \approx? \ (b\,a)\cdot M_7 \ , \ b \ \#? \ \langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} M_6 \approx? \ (b\,a)\cdot M_7 \ , \ b \ \#? \ \langle a, M_7\rangle$$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \;\approx?\; a.a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle b, M_6\rangle \approx? a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle b, M_6\rangle \approx? \langle b, (b\,a)\cdot M_7\rangle \;,\; b \,\#?\, \langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b \approx? b \;,\; M_6 \approx? (b\,a)\cdot M_7 \;,\; b \,\#?\, \langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} M_6 \approx? (b\,a)\cdot M_7 \;,\; b \,\#?\, \langle a, M_7\rangle$$

$$\stackrel{[M_6:=(b\,a)\cdot M_7]}{\Longrightarrow} b \,\#?\, \langle a, M_7\rangle$$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \;\approx?\; a.a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle b, M_6\rangle \approx? a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle b, M_6\rangle \approx? \langle b, (b\,a)\cdot M_7\rangle \;,\; b \;\#?\; \langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b \approx? b \;,\; M_6 \approx? (b\,a)\cdot M_7 \;,\; b \;\#?\; \langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} M_6 \approx? (b\,a)\cdot M_7 \;,\; b \;\#?\; \langle a, M_7\rangle$$

$$\stackrel{[M_6:=(b\,a)\cdot M_7]}{\Longrightarrow} b \;\#?\; \langle a, M_7\rangle$$

$$\stackrel{\varnothing}{\Longrightarrow} b \;\#?\; a \;,\; b \;\#?\; M_7$$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \;\approx?\; a.a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle b, M_6\rangle \approx? a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle b, M_6\rangle \approx? \langle b, (b\,a)\cdot M_7\rangle \;,\; b \;\#?\; \langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b \approx? b \;,\; M_6 \approx? (b\,a)\cdot M_7 \;,\; b \;\#?\; \langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} M_6 \approx? (b\,a)\cdot M_7 \;,\; b \;\#?\; \langle a, M_7\rangle$$

$$\stackrel{[M_6:=(b\,a)\cdot M_7]}{\Longrightarrow} b \;\#?\; \langle a, M_7\rangle$$

$$\stackrel{\varnothing}{\Longrightarrow} b \;\#?\; a \;,\; b \;\#?\; M_7$$

$$\stackrel{\varnothing}{\Longrightarrow} b \;\#?\; M_7$$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \;\approx?\; a.a.\langle a, M_7\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} b.\langle b, M_6\rangle \approx? a.\langle a, M_7\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} \langle b, M_6\rangle \approx? \langle b, (b\,a)\cdot M_7\rangle \;,\; b \#? \langle a, M_7\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} b \approx? b \;,\; M_6 \approx? (b\,a)\cdot M_7 \;,\; b \#? \langle a, M_7\rangle$$

$$\overset{\varepsilon}{\Longrightarrow} M_6 \approx? (b\,a)\cdot M_7 \;,\; b \#? \langle a, M_7\rangle$$

$$\overset{[M_6:=(b\,a)\cdot M_7]}{\Longrightarrow} b \#? \langle a, M_7\rangle$$

$$\overset{\varnothing}{\Longrightarrow} b \#? a \;,\; b \#? M_7$$

$$\overset{\varnothing}{\Longrightarrow} b \#? M_7$$

$$\overset{\{b\# M_7\}}{\Longrightarrow} \varnothing$$

Answers to the Quiz

$$a.b.\langle b, M_6\rangle \approx? \; a.a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} b.\langle b, M_6\rangle \approx? \; a.\langle a, M_7\rangle$$

$$\stackrel{\varepsilon}{\Longrightarrow} \langle b, M_6\rangle \approx ?$$

$$\stackrel{\varepsilon}{\Longrightarrow} b \approx? \; b \;,\; M$$

$$\stackrel{\varepsilon}{\Longrightarrow} M_6 \approx? \; (b$$

$$\stackrel{[M_6:=(b\,a)\cdot M_7]}{\Longrightarrow} \quad b \; \#?$$

$$\stackrel{\varnothing}{\Longrightarrow} b \; \#? \; a \;,\; b \; \#? \; M_7$$

$$\stackrel{\varnothing}{\Longrightarrow} b \; \#? \; M_7$$

$$\stackrel{\{b\#M_7\}}{\Longrightarrow} \varnothing$$

$$\lambda a.\lambda b.(b\, M_6) \;=_\alpha\; \lambda a.\lambda a.(a\, M_7)$$

we can take M_7 to be any λ-term that does not contain free occurrences of b, so long as we take M_6 to be the result of swapping all occurrences of b and a throughout M_7

Conclusion

- used a permutation operation for renaming (has much nicer properties)

Conclusion

- used a permutation operation for renaming (has much nicer properties) !!!

Conclusion

- used a permutation operation for renaming (has much nicer properties) !!!
- have concrete names for binders (nominal unification) and **not** de-Bruijn indices

Conclusion

- used a permutation operation for renaming (has much nicer properties) !!!
- have concrete names for binders (nominal unification) and **not** de-Bruijn indices
- it is a completely first-order language

Conclusion

- used a permutation operation for renaming (has much nicer properties) !!!
- have concrete names for binders (nominal unification) and **not** de-Bruijn indices
- it is a completely first-order language
- computed with freshness assumptions; this allowed us to define $\approx$ so that substitution respects α-equivalence

Conclusion

- used a permutation operation for renaming (has much nicer properties) !!!
- have concrete names for binders (nominal unification) and **not** de-Bruijn indices
- it is a completely first-order language
- computed with freshness assumptions; this allowed us to define $\approx$ so that substitution respects α-equivalence
- verified everything in Isabelle

Is it useful?

applications to logic programming (with J. Cheney)

$$\frac{x : A \in \Gamma}{\Gamma \triangleright x : A} \qquad \frac{\Gamma \triangleright M : A \supset B \quad \Gamma \triangleright N : A}{\Gamma \triangleright M\,N : B} \qquad \frac{x : A, \Gamma \triangleright M : B}{\Gamma \triangleright \lambda x.M : A \supset B}$$

```
type Gamma (var X) A :- member (pair X A) Gamma.

type Gamma (app M N) B :- type Gamma M (arrow A B),
                          type Gamma N A.

type Gamma (lam x.M) (arrow A B) / x#Gamma :-
                          type (pair x A)::Gamma M B.

member A A::Tail.

member A B::Tail :- member A Tail.
```

Is it useful?

applications to logic programming (with J. Cheney)

$$\frac{x{:}A \in \Gamma}{\Gamma \triangleright x{:}A} \qquad \frac{\Gamma \triangleright M{:}A \supset B \quad \Gamma \triangleright N{:}A}{\ldots} \qquad \frac{x{:}A,\Gamma \triangleright M{:}B}{\ldots \supset B}$$

αProlog is available from

```
www.cs.cornell.edu
  /people/jcheney/aprolog/
```

```
type Gamm

type Gamm                                             ),
                                   type Gamma N A.

type Gamma (lam x.M) (arrow A B) / x#Gamma :-
                                   type (pair x A)::Gamma M B.

member A A::Tail.

member A B::Tail :- member A Tail.
```

Future Work: Nominal Logic

Wouldn't it be nice to have an (intelligible) first-order logic for reasoning about syntax involving meta-variables and binders? Instead of the usual axioms

$$\frac{}{P(t), \Gamma \vdash \Delta, P(t)}\ \text{axiom}$$

one would have axioms of the form

$$\frac{\nabla \vdash t_1 \approx t_2}{\nabla; P(t_1), \Gamma \vdash \Delta, P(t_2)}\ \text{axiom}$$

where nominal terms are treated 'modulo $\approx$'.
Goal: easy induction principles, meta-vars,. . .

(Related work is $FO\lambda^{\Delta I\!N}$ by McDowell & Miller.)

The End

Paper, implementation and Isabelle scripts at:

www.cl.cam.ac.uk/~cu200/Unification