Nominal Logic and Reasoning about α-Equivalent Syntax

Christian Urban

every Tuesday from
14:15 to 16:00 in Room E27

(except 26. April, 17. May and 5. July)

Do You 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?

First-Order Unification

- $\mathtt{f}(t_1, \ldots, t_n) \approx? \mathtt{f}(s_1, \ldots, s_n) \cup P$

  $\Longrightarrow t_1 \approx? s_1 \cup \ldots \cup t_n \approx? s_n \cup P$

- $X \approx? t \cup P$

  $\overset{[X:=t]}{\Longrightarrow} P[X := t]$

  $[X := t]$ part of the answer

- ...

- if σ is an answer for a problem P, then σ is the most general solution

First-Order Unification

- $\mathtt{f}(t_1, \ldots, t_n) \approx? \mathtt{f}(s_1, \ldots, s_n) \cup P$

  $\Longrightarrow t_1 \approx? s_1 \cup \ldots \cup t_n \approx? s_n \cup P$

- $X \approx? t \cup P$

  $[X := t]$ $\Longrightarrow$ part of the answer

In this lecture I shall give a **slightly** more complicated algorithm for unification modulo α.

- ...

- if σ is an answer for a problem P, then σ is the most general solution

Simple Problem

Given the inference rule:

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

Simple Problem

Given the inference rule:

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

$\mathtt{fn}\ a.(-)$ and $\mathtt{let}\ a = X\ \mathtt{in}\ (-)$ bind all occurrences of a in $(-)$

Simple Problem

Given the inference rule:

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

- $\texttt{let}\, a = 1\, \texttt{in}\, a \Downarrow 1$

Simple Problem

Given the inference rule:

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

- $\texttt{let}\ a = 1\ \texttt{in}\ a \Downarrow 1\ \ [Y := a; X, V := 1]$

Simple Problem

Given the inference rule:

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

error!

- $\texttt{let}\ a = 1\ \texttt{in}\ a \Downarrow 1 \quad [Y := a; X, V := 1]$
- $\texttt{let}\ b = 1\ \texttt{in}\ b \Downarrow 1 \quad [Y := b; X, V := 1]$

Simple Problem

Given the inference rule:

$$\frac{\texttt{app}(\texttt{fn}\ \lambda a.Fa)\ X\ \Downarrow\ V}{\texttt{let}\ X(\lambda a.Fa)\ \Downarrow\ V}$$

Simple Problem

Given the inference rule:

$$\frac{\texttt{app}(\texttt{fn}\ F)\ X \ \Downarrow\ V}{\texttt{let}\ X\ F \ \Downarrow\ V}$$

- $\texttt{let}\ 1\ \lambda a.a \Downarrow 1$ or $\texttt{let}\ 1\ \lambda b.b \Downarrow 1$

Simple Problem

Given the inference rule:

$$\frac{\texttt{app}(\texttt{fn}\ F)\ X \Downarrow V}{\texttt{let}\ X\ F \Downarrow V}$$

- $\texttt{let}\ 1\ \lambda a.a \Downarrow 1$ or $\texttt{let}\ 1\ \lambda b.b \Downarrow 1$

Does it have to be so complicated? No!

Recap α-Equivalence

$$\frac{}{a \approx a}\ \approx\text{-atm}$$

$$\frac{t_1 \approx s_1 \quad t_2 \approx s_2}{t_1\, t_2 \approx s_1\, s_2}\ \approx\text{-app}$$

$$\frac{t \approx s}{\lambda a.t \approx \lambda a.s}\ \approx\text{-lam}_1$$

$$\frac{t \approx (a\,b)\bullet s \quad a \# s}{\lambda a.t \approx \lambda b.s}\ \approx\text{-lam}_2$$

$$\frac{}{a \# b}\ \#\text{-atm}$$

$$\frac{a \# t_1 \quad a \# t_2}{a \# t_1\, t_2}\ \#\text{-app}$$

$$\frac{}{a \# \lambda a.t}\ \#\text{-lam}_1$$

$$\frac{a \# t}{a \# \lambda b.t}\ \#\text{-lam}_2$$

assuming $a \neq b$

Permutations on λ-Terms

$$[]\bullet a \stackrel{\text{def}}{=} a$$

$$((a_1\, a_2) :: \pi)\bullet a \stackrel{\text{def}}{=} \begin{cases} a_1 & \text{if } \pi\bullet a = a_2 \\ a_2 & \text{if } \pi\bullet a = a_1 \\ \pi\bullet a & \text{otherwise} \end{cases}$$

$$\pi\bullet (t_1\, t_2) \stackrel{\text{def}}{=} (\pi\bullet t_1)(\pi\bullet t_2)$$

$$\pi\bullet(\lambda a.t) \stackrel{\text{def}}{=} \lambda(\pi\bullet a).(\pi\bullet t)$$

Problems from the Quiz

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

What to do with permutations on 'holes' ?

$\pi \bullet \square$

Problems from the Quiz

- $\lambda a.\lambda b.(\Box_1\ b) \approx? \lambda b.\lambda a.(a\ \Box_1)$
- $\lambda a.\lambda b.(\Box_2\ b) \approx? \lambda b.\lambda a.(a\ \Box_3)$
- $\lambda a.\lambda b.(b\ \Box_4) \approx? \lambda b.\lambda a.(a\ \Box_5)$
- $\lambda a.\lambda b.(b\ \Box_6) \approx? \lambda a.\lambda a.(a\ \Box_7)$

What to do with permutations on 'holes' ?

$$\pi \bullet \Box$$

Answer: Nothing! Just keep them.

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:

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

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:

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

Permutations

a permutation applied to a term:

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

inverse permutation given by reversing the list of swappings

Permutations on atoms are bijections!

$$\pi \bullet a = b \quad \text{iff} \quad a = (\pi^{-1}) \bullet b$$

Freshness Relation

We will identify

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

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

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

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

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

Freshness Relation

We will identify

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

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

$$t \approx t'$$ α-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

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

$$\nabla \vdash a \# t \qquad \text{freshness}$$

where

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

is a finite set of freshness assumptions.

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

Rules for Equivalence

Excerpt
(i.e. only the interesting rules)

Rules for Equivalence

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

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

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

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

$$\frac{}{\nabla \vdash a \# a.t}$$

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

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

$\approx$ 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 $\varnothing \vdash t \approx t'$
>
> $a \notin FA(t)$ 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

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

Comparison with $=_\alpha$

That is a crucial point: if we had

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

then applying $[X := a], [X := b], \ldots$
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.

f

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

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

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

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

$$\Rightarrow \;\; a.\langle a, (a\,b)\cdot Y\rangle$$

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 $\nabla \vdash a \# \sigma(t)$.

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

- $\nabla_2 \vdash \sigma(\nabla_1)$

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

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

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

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

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

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.

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.

straightforward definition:
"iff there exists a τ such that ..."

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).

Reductions

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

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

Reductions

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

Reductions

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

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

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

  $\{t \approx? \; (a\,b) \bullet t', a \; \#? \; t'\} \cup P$

- $\{a.t \approx? \; a.t'\} \uplus P \overset{\varepsilon}{\Longrightarrow} \{t \approx? \; t'\} \cup P$

- $\{\pi \bullet X \approx? \; \pi' \bullet X\}$

  $\{a \; \#? \; X \mid a \in \mathit{ds}(\pi, \pi')\} \cup P$

  $\sigma = [X := \pi^{-1} \bullet t]$

- $\{\pi \bullet X \approx? \; t\} \uplus P \overset{\sigma}{\Longrightarrow} \sigma P$

  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

- $\{a.t \approx? \; b.t'\} \uplus P \stackrel{\varepsilon}{\Longrightarrow}$ 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 $a \neq b$
- $\{a \; \#? \; \pi \bullet X\} \uplus P \stackrel{\nabla}{\Longrightarrow} P$

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
  $\nabla = \{\pi^{-1} \bullet a \# X\}$
- $\{a \#? \; b.t\} \uplus P \stackrel{\varnothing}{\Longrightarrow} \{a \#? \; t\} \cup P$ if $a \neq b$
- $\{a \#? \; \pi \bullet X\} \uplus P \stackrel{\nabla}{\Longrightarrow} P$

Reductions

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

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

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.

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

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

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

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

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

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

$$\Longrightarrow FAIL$$

$\lambda a.\lambda b.(M_1\, b) =_\alpha \lambda b.\lambda a.(a\, M_1)$ 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Answers to the Quiz

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

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

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

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

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

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

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

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

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

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

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

Correct/Incorrect?

Does the following Prolog program produce for every lambda-term the correct type?

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

```
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) :-
                          type (pair X A)::Gamma M B.

member A A::Tail.

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

Correct/Incorrect?

NO! The term $\lambda x.\lambda x.(x\ x)$ should not be ev-typable. But

```
type [] (lam X (lam X (app X X))) T
```

returns two answers

- $\tau \supset (\tau \supset \sigma) \supset \sigma$
- $(\tau \supset \sigma) \supset \tau \supset \sigma$

```
                                    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)

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

```
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.
```

Conclusion

- computed with freshness assumptions. This allowed us to define $\approx$ so that substitution respects α-equivalence. Furthermore have concrete names for binders (nominal unification) and not de-Bruijn indices.
- it is a completely first-order language
- verified everything in Isabelle
- as powerful as Miller's pattern-unification
- algorithm is quadratic (linear???)

Possible Extensions

- Atoms are constants: otherwise

$$a.M \approx? \; X.Y.\mathtt{f}(a)$$

  - 1st solution: $[M := X.\mathtt{f}(X), Y := a]$

$$a.X.\mathtt{f}(X) \approx X.a.\mathtt{f}(a)$$

  - 2nd solution: $[M := Y.\mathtt{f}(a), X := a]$

$$a.Y.\mathtt{f}(a) \approx a.Y.\mathtt{f}(a)$$