Unification Modulo Observational Equivalence

over simply-typed λ-terms in call-by-value semantics

Stéphane Gimenez, Joe Wells

18/08/04

Motivations

- C. Haack proposed a tool for automatic adaptation of software components that would need UMOE.
  - The approximation made was to use HOU to find unification candidates modulo β-equivalence, then check in a second time that the observational behavior are the same.
- We propose to find solutions in a single phase.
  - Possibly, finding solutions that are not needed to respect β-equivalence.

Unification

- Classical unification problems deals with solving equations at the syntax level modulo some equivalence relations such as associativity or commutativity.

$$b + X + Y \approx a + Z$$

Unification

- Classical unification problems deals with solving equations at the syntax level modulo some equivalence relations such as associativity or commutativity.

$$b + X + Y \approx a + Z$$

- Ground Solution:

$$X \mapsto a,\ Y \mapsto a,\ Z \mapsto b + a$$

Unification

- Classical unification problems deals with solving equations at the syntax level modulo some equivalence relations such as associativity or commutativity.

$$b + X + Y \approx a + Z$$

- Ground Solution:

$$X \mapsto a,\ Y \mapsto a,\ Z \mapsto b + a$$

- Unifiers:

$$Y \mapsto a,\ Z \mapsto b + X$$

$$Y \mapsto a + T,\ Z \mapsto b + T + X$$

Definition

An unification problem is a set of equations $t_1 \approx t_2$ in an algebra extended with unknowns $X, Y, Z \ldots$, for which equivalence is written $\simeq$.

Definition

An unification problem is a set of equations $t_1 \approx t_2$ in an algebra extended with unknowns $X, Y, Z \ldots$, for which equivalence is written $\simeq$.

Definition

An unifier for a given unification problem is a substitution θ (that replaces unknowns with terms) such for each equation $t_1 \approx t_2$ of the unification problem, $\theta t_1 \simeq \theta t_2$.

Definition

An unification problem is a set of equations $t_1 \approx t_2$ in an algebra extended with unknowns $X, Y, Z...$, for which equivalence is written $\simeq$.

Definition

An unifier for a given unification problem is a substitution θ (that replaces unknowns with terms) such for each equation $t_1 \approx t_2$ of the unification problem, $\theta t_1 \simeq \theta t_2$.

Definition

An unifier θ_1 is said more general than θ_2 ($\theta_1 \leq \theta_2$) iff there exists a substitution θ such that $\theta_2 = \theta\theta_1$

- In our example,

$$X \mapsto a,\ Y \mapsto a,\ Z \mapsto b + a$$

$$\geq Y \mapsto a,\ Z \mapsto b + X$$

$$\geq Y \mapsto a + T,\ Z \mapsto b + T + X$$

- In our example,

$$X \mapsto a,\ Y \mapsto a,\ Z \mapsto b + a$$

$$\geq Y \mapsto a,\ Z \mapsto b + X$$

$$\geq Y \mapsto a + T,\ Z \mapsto b + T + X$$

- In fact there are two minimal unifiers,

$$X \mapsto a + T,\ Z \mapsto b + T + Y$$

$$Y \mapsto a + T,\ Z \mapsto b + T + X$$

Objectives

- Find a most general unifier when it exists.
- Find a complete finite (finitely representable) set of minimal unifiers.
- Find a complete finite (finitely representable) set of unifiers.
- Enumerate a complete set of unifiers.
- Find an unifier when there is one.

The existence of an unifier is undecidable for almost every "complex" algebra, only the two last specifications can be assured. Incomplete results can also be interesting.

Higher Order

- When the algebra considered is the algebra of λ-terms modulo $\beta\eta$-equivalence, unification is said Higher Order Unification.
- Higher Order Unification is semi-decidable.
- A exhaustive "generate and test" algorithm allows to know that a specific problem has solutions.
- Huet's algorithms allows to restrict the search space.

Definition

Simply-typed λ-terms are built using the following syntax:

$$
\begin{array}{rcl}
l^{\sigma \to \tau} & ::= & \lambda x^{\sigma} . \, t^{\tau} \\
t^{\tau} & ::= & l^{\tau} \;\mid\; x^{\tau} \;\mid\; X^{\tau} \;\mid\; t_1^{\sigma \to \tau} \, t_2^{\sigma}
\end{array}
$$

Definition

A unification problem is syntactically defined as:

$$
P \quad ::= \quad P_1, P_2 \;\mid\; t_1^{\tau} \approx t_2^{\tau} \;\mid\; \varnothing \;\mid\; \bot
$$

Huet's Algorithm

Rules for Higher Order Unification

- delete:

$$P, t \approx t \;\rightarrow\; P$$

- decompose:

$$P, x\,\vec{t} \approx x\,\vec{t}' \;\rightarrow\; P, t_1 \approx t_1', \ldots, t_n \approx t_n'$$

- eliminate:

$$P, X \approx t \;\rightarrow\; P[X := t], X \approx t \quad \text{if } X \notin \text{fv}\, t$$

- imitate: ω ranges over x and X,

$$P, X\,\vec{t} \approx \omega\,\vec{t'} \quad \rightarrow \quad P, X\,\vec{t} \approx \omega\,\vec{t'},$$
$$X = \lambda\vec{r}.\,\omega(\lambda\vec{s}_1.\,Z_1(\vec{r},\vec{s}_1), \ldots, \lambda\vec{s}_n.\,Z_n(\vec{r},\vec{s}_n))$$

- project:

$$P, X \approx x\,\vec{t} \quad \rightarrow \quad P, X \approx x\,\vec{t},$$
$$X = \lambda\vec{r}.\,r_i(\lambda\vec{s}_1.\,Z_1(\vec{r},\vec{s}_1), \ldots, \lambda\vec{s}_n.\,Z_n(\vec{r},\vec{s}_n))$$

- guess:

$$P, X\,\vec{t} \approx Y\,\vec{t'} \quad \rightarrow \quad P, X\,\vec{t} \approx Y\,\vec{t'},$$
$$X = \lambda\vec{r}.\,\omega(\lambda\vec{s}_1.\,Z_1(\vec{r},\vec{s}_1), \ldots, \lambda\vec{s}_n.\,Z_n(\vec{r},\vec{s}_n))$$

Unification Modulo Observational Equivalence

- A different kind of unification on simply-typed λ-terms.
- Observational equivalence instead of $\beta\eta$-equivalence.
- Call-by-value semantics, since the two equivalences are the same in call-by-name semantics.

Calculus

Definition

Simply-typed λ-terms are built using the following syntax:

$$
\begin{array}{rcl}
l^{\sigma \to \tau} & ::= & \lambda x^{\sigma} . \, t^{\tau} \\
v^{\tau} & ::= & l^{\tau} \;\big|\; x^{\tau} \;\big|\; \underline{X}^{\tau}_{\Gamma} \\
t^{\tau} & ::= & v^{\tau} \;\big|\; t_1^{\sigma \to \tau} \; t_2^{\sigma} \;\big|\; \bar{X}^{\tau}_{\Gamma} \\
\Gamma & ::= & t^{\tau}, \Gamma \;\big|\; \varnothing
\end{array}
$$

Calculus

Definition

Simply-typed λ-terms are built using the following syntax:

$$
\begin{array}{rcl}
l^{\sigma \to \tau} & ::= & \lambda x^{\sigma} . \, t^{\tau} \\
v^{\tau} & ::= & l^{\tau} \;\mid\; x^{\tau} \;\mid\; \underline{X}^{\tau}_{\Gamma} \\
t^{\tau} & ::= & v^{\tau} \;\mid\; t_1^{\sigma \to \tau} \; t_2^{\sigma} \;\mid\; \bar{X}^{\tau}_{\Gamma} \\
\Gamma & ::= & t^{\tau}, \Gamma \;\mid\; \varnothing \\
 & & \\
X^{\tau}_{\Gamma} & ::= & \underline{X}^{\tau}_{\Gamma} \;\mid\; \bar{X}^{\tau}_{\Gamma}
\end{array}
$$

Definition

The inferior bound set of *free variables* $\mathrm{fv}_{\mathrm{inf}}\ t$ of a term t is defined according to the following rules:

- usual rules:

$$\begin{aligned} \mathrm{fv}_{\mathrm{inf}}\ \lambda z^{\sigma}.\ t^{\tau} &= \mathrm{fv}_{\mathrm{inf}}\ t^{\tau} \setminus \{z^{\sigma}\} \\ \mathrm{fv}_{\mathrm{inf}}\ x^{\tau} &= \{x^{\tau}\} \\ \mathrm{fv}_{\mathrm{inf}}\ t_1^{\sigma\to\tau}\ t_2^{\sigma} &= \mathrm{fv}_{\mathrm{inf}}\ t_1^{\sigma\to\tau} \cup \mathrm{fv}_{\mathrm{inf}}\ t_2^{\sigma} \end{aligned}$$

Definition

The inferior bound set of *free variables* $\mathrm{fv}_{\mathrm{inf}}\ t$ of a term t is defined according to the following rules:

- usual rules:

$$\begin{aligned} \mathrm{fv}_{\mathrm{inf}}\ \lambda z^{\sigma}.\, t^{\tau} &= \mathrm{fv}_{\mathrm{inf}}\ t^{\tau} \setminus \{z^{\sigma}\} \\ \mathrm{fv}_{\mathrm{inf}}\ x^{\tau} &= \{x^{\tau}\} \\ \mathrm{fv}_{\mathrm{inf}}\ t_1^{\sigma\rightarrow\tau}\ t_2^{\sigma} &= \mathrm{fv}_{\mathrm{inf}}\ t_1^{\sigma\rightarrow\tau} \cup \mathrm{fv}_{\mathrm{inf}}\ t_2^{\sigma} \end{aligned}$$

- extended with:

$$\mathrm{fv}_{\mathrm{inf}}\ X_{\Gamma}^{\tau} = \emptyset$$

Definition

The superior bound set of *free variables* $\mathrm{fv}_{\mathrm{sup}}\, t$ of a term t is defined according to the following rules:

- usual rules:

$$
\begin{aligned}
\mathrm{fv}_{\mathrm{sup}}\, \lambda z^{\sigma}.\, t^{\tau} &= \mathrm{fv}_{\mathrm{sup}}\, t^{\tau} \setminus \{z^{\sigma}\} \\
\mathrm{fv}_{\mathrm{sup}}\, x^{\tau} &= \{x^{\tau}\} \\
\mathrm{fv}_{\mathrm{sup}}\, t_1^{\sigma \to \tau}\ t_2^{\sigma} &= \mathrm{fv}_{\mathrm{sup}}\, t_1^{\sigma \to \tau} \cup \mathrm{fv}_{\mathrm{sup}}\, t_2^{\sigma}
\end{aligned}
$$

Definition

The superior bound set of *free variables* $\mathrm{fv}_{\mathrm{sup}}\, t$ of a term t is defined according to the following rules:

- usual rules:

$$\begin{aligned} \mathrm{fv}_{\mathrm{sup}}\, \lambda z^{\sigma}.\, t^{\tau} &= \mathrm{fv}_{\mathrm{sup}}\, t^{\tau} \setminus \{z^{\sigma}\} \\ \mathrm{fv}_{\mathrm{sup}}\, x^{\tau} &= \{x^{\tau}\} \\ \mathrm{fv}_{\mathrm{sup}}\, t_1^{\sigma\to\tau}\ t_2^{\sigma} &= \mathrm{fv}_{\mathrm{sup}}\, t_1^{\sigma\to\tau} \cup \mathrm{fv}_{\mathrm{sup}}\, t_2^{\sigma} \end{aligned}$$

- extended with:

$$\mathrm{fv}_{\mathrm{sup}}\, X_{\Gamma}^{\tau} = \bigcup_{t^{\sigma}\in\Gamma} \mathrm{fv}_{\mathrm{sup}}\, t^{\sigma}$$

Definition

A *substitution operator for variables* is a pair $[x^\tau := t^\tau]$.

- usual rules:

$$
\begin{array}{rcl}
x^\tau[x^\tau := t^\tau] & = & t^\tau \\
y^\sigma[x^\tau := t^\tau] & = & y^\sigma \quad \text{if } x^\tau \neq y^\sigma \\
(t_1\ t_2)[x^\tau := t^\tau] & = & t_1[x^\tau := t^\tau]\ t_2[x^\tau := t^\tau] \\
(\lambda z^\sigma .\, t')[x^\tau := t^\tau] & = & \lambda z^\sigma .\, t'[x^\tau := t^\tau] \quad \text{if } \begin{cases} z^\sigma \neq x^\tau \\ z^\sigma \notin \mathrm{fv}_{\mathrm{sup}}\, t \end{cases}
\end{array}
$$

Definition

A *substitution operator for variables* is a pair $[x^\tau := t^\tau]$.

- usual rules:

$$\begin{array}{rcl} x^\tau[x^\tau := t^\tau] & = & t^\tau \\ y^\sigma[x^\tau := t^\tau] & = & y^\sigma \quad \text{if } x^\tau \neq y^\sigma \\ (t_1\, t_2)[x^\tau := t^\tau] & = & t_1[x^\tau := t^\tau]\, t_2[x^\tau := t^\tau] \\ (\lambda z^\sigma .\, t')[x^\tau := t^\tau] & = & \lambda z^\sigma .\, t'[x^\tau := t^\tau] \quad \text{if } \begin{cases} z^\sigma \neq x^\tau \\ z^\sigma \notin \mathrm{fv}_{\mathrm{sup}}\, t \end{cases} \end{array}$$

- extended with:

$$X^\sigma_\Gamma[x^\tau := t^\tau] = X^\sigma_{\Gamma[x^\tau := t^\sigma]}$$

- where:

$$\begin{array}{rcl} (t'^\sigma, \Gamma)[x^\tau := t^\tau] & = & t'^\sigma[x^\tau := t^\tau], \Gamma[x^\tau := t^\tau] \\ \varnothing[x^\tau := t^\sigma] & = & \varnothing \end{array}$$

Definition

A *substitution operator for unknowns* is a pair $[X_\Sigma^\tau := t^\tau]$, where Σ is a vector of distinct variables and t a term which does not contain X, such that $\mathrm{fv}_{\mathrm{sup}}\, t^\tau \subseteq \Sigma$, defined modulo α-conversion of the variables in Γ.

- transition rules:

$$\begin{array}{rcl} (\lambda z^\sigma.\, t')[X_\Sigma^\tau := t^\tau] & = & \lambda z^\sigma.\, t'[X_\Sigma^\tau := t^\tau] \\ (t_1^{\sigma\to\tau}\ t_2^\sigma)[X_\Sigma^\tau := t^\tau] & = & t_1^{\sigma\to\tau}[X_\Sigma^\tau := t^\tau]\ t_2^\sigma[X_\Sigma^\tau := t^\tau] \\ x^\tau[X_\Sigma^\tau := t^\tau] & = & x^\tau \end{array}$$

- unknowns replacement:

$$
\begin{aligned}
Y^{\tau}_{\Gamma}[X^{\tau}_{\Sigma} := t^{\tau}] &= Y^{\tau}_{\Gamma[X^{\tau}_{\Sigma} := t^{\tau}]} \quad \text{if } X \neq Y \\
X^{\tau}_{\Gamma}[X^{\tau}_{\Sigma} := t^{\tau}] &= t^{\tau}[\Sigma := \Gamma]
\end{aligned}
$$

- where:

$$
\begin{aligned}
t'[x^{\sigma}, \Sigma := t^{\tau}, \Gamma] &= t'[x^{\sigma} := t^{\tau}][\Sigma := \Gamma] \\
t'[\varnothing := \varnothing] &= t'
\end{aligned}
$$

Semantics

Definition

Call-by-value reduction is the smallest binary relation $\longrightarrow_v$ over λ-terms that satisfies:

$$\frac{t_1^{\sigma\to\tau} \longrightarrow_v t_2^{\sigma\to\tau}}{t_1^{\sigma\to\tau}\ t^{\sigma} \longrightarrow_v t_2^{\sigma\to\tau}\ t^{\sigma}}\,v_{\mathrm{left}} \qquad \frac{t_1^{\sigma} \longrightarrow_v t_2^{\sigma}}{v^{\sigma\to\tau}\ t_1^{\sigma} \longrightarrow_v v^{\sigma\to\tau}\ t_2^{\sigma}}\,v_{\mathrm{right}}$$

$$\frac{}{(\lambda x^{\sigma}.\, t^{\tau})\ v^{\sigma} \longrightarrow_v t^{\tau}[x^{\sigma} := v^{\sigma}]}\,v_{\beta}$$

Semantics

Definition

Call-by-value reduction is the smallest binary relation $\longrightarrow_v$ over λ-terms that satisfies:

$$\frac{t_1^{\sigma\rightarrow\tau} \longrightarrow_v t_2^{\sigma\rightarrow\tau}}{t_1^{\sigma\rightarrow\tau}\ t^{\sigma} \longrightarrow_v t_2^{\sigma\rightarrow\tau}\ t^{\sigma}}\,v_{\mathrm{left}} \qquad \frac{t_1^{\sigma} \longrightarrow_v t_2^{\sigma}}{v^{\sigma\rightarrow\tau}\ t_1^{\sigma} \longrightarrow_v v^{\sigma\rightarrow\tau}\ t_2^{\sigma}}\,v_{\mathrm{right}}$$

$$\frac{}{(\lambda x^{\sigma}.\, t^{\tau})\ v^{\sigma} \longrightarrow_v t^{\tau}[x^{\sigma} := v^{\sigma}]}\,v_{\beta}$$

$\underline{X}_\Gamma$ and $\bar{X}_\Gamma$ behave differently:

$$\bar{X}_\Gamma\,((\lambda z.\, z)\ u) \quad \underline{X}_\Gamma\,((\lambda z.\, z)\ u)$$

Definition

Evaluation context.

$$\begin{array}{lcl} c^{\sigma \Rightarrow \sigma} & ::= & \Box^{\sigma} \\ c^{\sigma \Rightarrow \tau} & ::= & c^{\sigma \Rightarrow \tau' \to \tau} \; t^{\tau'} \;\mid\; l^{\tau' \to \tau} \; c^{\sigma \Rightarrow \tau'} \end{array}$$

Lemma

Normal forms of type τ for the call-by-value semantics are exactly the terms of the form:

$$v^{\tau}$$

$$c^{\sigma \Rightarrow \tau}[x^{\sigma' \to \sigma} \; v^{\sigma'}]$$

$$c^{\sigma \Rightarrow \tau}[\underline{X}_{\Gamma}^{\sigma' \to \sigma} \; v^{\sigma'}]$$

$$c^{\sigma \Rightarrow \tau}[\bar{X}_{\Gamma}^{\sigma}]$$

Observational equivalence

Definition

A *congruence* for the simply-typed λ-calculus is a relation $\sim$ that satisfies:

$$\frac{t_1 \sim t_2}{\lambda x.\, t_1 \sim \lambda x.\, t_2}\mathrm{cong_{abs}} \qquad \frac{t_1 \sim t_2 \quad t_1' \sim t_2'}{t_1\; t_1' \sim t_2\; t_2'}\mathrm{cong_{app}}$$

Observational equivalence

Definition

A *congruence* for the simply-typed λ-calculus is a relation $\sim$ that satisfies:

$$\frac{t_1 \sim t_2}{\lambda x.\, t_1 \sim \lambda x.\, t_2}\mathrm{cong}_{\mathrm{abs}} \qquad \frac{t_1 \sim t_2 \quad t_1' \sim t_2'}{t_1\ t_1' \sim t_2\ t_2'}\mathrm{cong}_{\mathrm{app}}$$

$$\frac{t_1 \sim t_1', \ldots, t_n \sim t_n'}{X_{t_1,\ldots,t_n} \sim X_{t_1',\ldots,t_n'}}\mathrm{cong}_{\mathrm{scope}}$$

Definition

The *blocking symbol* $\neg\, t$ of a term t is defined according to the normal form of t, using the following matching:

$$
\begin{aligned}
\neg\, t = \cdot \quad &\iff \quad t{\downarrow_v} = v \\
\neg\, t = x \quad &\iff \quad t{\downarrow_v} = c[x\ v] \\
\neg\, t = \underline{X}_\Gamma \quad &\iff \quad t{\downarrow_v} = c[\underline{X}_\Gamma\ v] \\
\neg\, t = \bar{X}_\Gamma \quad &\iff \quad t{\downarrow_v} = c[\bar{X}_\Gamma]
\end{aligned}
$$

- The blocking symbol plays the same role as the head variable in HOU.

Definition

A *bisimulation* is a congruence $\sim$ such that:

$$t_1 \sim t_2 \;\Rightarrow\; \neg\, t_1 = \neg\, t_2$$

$$\frac{t_1 \sim t_2}{t_1{\downarrow_v} \sim t_2{\downarrow_v}}\text{eval}$$

Definition

The *observational equivalence* $\simeq$ is the greatest bisimulation. It exists, because the union of two bisimulations is also a bisimulation.

Example

- The booleans can be distinguished.

$$\lambda x^{\gamma}.\, \lambda y^{\gamma}.\, x^{\gamma} \quad \not\simeq \quad \lambda x^{\gamma}.\, \lambda y^{\gamma}.\, y^{\gamma}$$

- η-equivalent terms are not necessarily observationally equivalent,

$$f^{\sigma \to \tau \to \tau'}\, x^{\sigma} \quad \not\simeq \quad \lambda z^{\tau}.\, f^{\sigma \to \tau \to \tau'}\, x^{\sigma}\, z^{\tau}$$

- unless the term is a value.

$$\lambda z^{\tau}.\, f^{\sigma \to \tau \to \tau'}\, x^{\sigma}\, z^{\tau} \quad \simeq \quad \lambda z'^{\tau}.\, (\lambda z^{\tau}.\, f^{\sigma \to \tau \to \tau'}\, x^{\sigma}\, z^{\tau})\, z'^{\tau}$$

Example

- Sometimes, β-equivalent terms are observationally equal in call-by-value semantics,

$$\lambda y^{\alpha}.\,(\lambda z^{\beta\to\beta}.\,y^{\alpha})\,(\lambda x^{\beta}.\,x^{\beta}) \;\simeq\; \lambda y^{\alpha}.\,y^{\alpha}$$

- Sometimes, not.

$$\lambda y^{\alpha}.\,(\lambda z^{\beta}.\,y^{\alpha})\,(f^{\gamma\to\beta}x^{\gamma}) \;\not\simeq\; \lambda y^{\alpha}.\,y^{\alpha}$$

Unification

Definition

A unification problem is syntactically defined as:

$$P \quad ::= \quad P_1, P_2 \quad | \quad t_1^\tau \approx t_2^\tau \quad | \quad \varnothing \quad | \quad \bot \quad | \quad (\nu x)\ P \quad | \quad (\nu X)\ P$$

Example

$$G_{f^{\alpha\to\alpha\to\alpha}}^{\alpha\to\alpha\to\alpha}\ x^\alpha\ y^\alpha \approx f^{\alpha\to\alpha\to\alpha}\ y^\alpha\ x^\alpha$$

unifier:

$$G_{f^{\alpha\to\alpha\to\alpha}}^{\alpha\to\alpha\to\alpha} \mapsto \lambda u^\alpha .\, \lambda v^\alpha .\, f^{\alpha\to\alpha\to\alpha}\ v^\alpha\ u^\alpha$$

Definition

A *unifier* for a given unification problem is a substitution whose domain is the set of unknowns of the problem that makes observationally equivalent the two terms of each disagreement pair.

Definition

A *unifier* for a given unification problem is a substitution whose domain is the set of unknowns of the problem that makes observationally equivalent the two terms of each disagreement pair.

The substitution term for an unknown must only use variables that appear as index of the unknown. Then the following substitution is not a candidate for being an unifier:

$$G^{\alpha\to\alpha\to\alpha}_{f^{\alpha\to\alpha\to\alpha}} \mapsto \lambda u^{\alpha}.\,\lambda v^{\alpha}.\,f^{\alpha\to\alpha\to\alpha}\;v^{\alpha}\;x^{\alpha}$$

Definition

A *unifier* for a given unification problem is a substitution whose domain is the set of unknowns of the problem that makes observationally equivalent the two terms of each disagreement pair.

The substitution term for an unknown must only use variables that appear as index of the unknown. Then the following substitution is not a candidate for being an unifier:

$$G^{\alpha\to\alpha\to\alpha}_{f^{\alpha\to\alpha\to\alpha}} \mapsto \lambda u^{\alpha}.\,\lambda v^{\alpha}.\,f^{\alpha\to\alpha\to\alpha}\;v^{\alpha}\;x^{\alpha}$$

Definition

Assuming we only need one representant by equivalence class, we restrict our interest space to unifiers whose right-sides are normal forms.

Solution

Towards a solving procedure...

Reusing HOU

The general principle of HOU can be reused: Instantiating unknowns, using the restrictions that can be grabbed using the equivalences already discovered.
But,

- the range of normal forms in call-by-value semantics is wider than for $\beta\eta$-reduction.
- normalization is not sufficient to know if two terms are equivalent.
- we cannot use β-reduction to deal with scope issues.

Augmented call-by-value reduction

Lemma

Assuming $z \notin \mathrm{fv}\, t_2$, $\quad (\lambda z^\sigma .\, t_1^{\tau' \to \tau})\, t^\sigma\, t_2^{\tau'} \simeq (\lambda z^\sigma .\, t_1^{\tau' \to \tau}\, t_2^{\tau'})\, t^\sigma$

Lemma

Assuming $z \notin \mathrm{fv}\, v$, $\quad v^{\tau \to \tau'}\, ((\lambda z^\sigma .\, t_1^\tau)\, t^\sigma) \simeq (\lambda z^\sigma .\, v^{\tau \to \tau'}\, t_1^\tau)\, t^\sigma$

Lemma

$(\lambda z^\sigma .\, z^\sigma)\, t^\sigma \simeq t^\sigma$

These are remarkable equivalences that are also β-equivalences.

Definition

Call-by-value augmented evaluation is defined according to:

$$
\begin{array}{rcll}
t{\downarrow_a} & = & (t{\downarrow_v}){\downarrow_a} & \text{if t is not a normal form} \\
x{\downarrow_a} & = & x & \\
(\lambda z.\, t){\downarrow_a} & = & \lambda z.\, t{\downarrow_a} & \\
c[x\, v]{\downarrow_a} & = & (\lambda w.\, c[w]{\downarrow_a})\, (x\, v{\downarrow_a}) & \\
X_\Gamma{\downarrow_a} & = & X_{\Gamma{\downarrow_a}} & \\
c[\underline{X}_\Gamma\, v]{\downarrow_a} & = & (\lambda w.\, c[w]{\downarrow_a})\, (\underline{X}_{\Gamma{\downarrow_a}}\, v{\downarrow_a}) & \\
c[\bar{X}_\Gamma]{\downarrow_a} & = & (\lambda w.\, c[w]{\downarrow_a})\, \bar{X}_{\Gamma{\downarrow_a}} &
\end{array}
$$

with:

$$
\begin{array}{rcl}
t, \Gamma{\downarrow_a} & = & t{\downarrow_a}, \Gamma{\downarrow_a} \\
\varnothing{\downarrow_a} & = & \varnothing
\end{array}
$$

Lemma

Normal forms for the augmented evaluation are exactly the terms of the form:

$$
\begin{array}{lcl}
m & ::= & x \\
 & | & \lambda z.\, m \\
 & | & (\lambda w.\, m)\,(x\; m) \\
 & | & X_{\vec{m}} \\
 & | & (\lambda w.\, m)\,(\underline{X}_{\vec{m}}\; m) \\
 & | & (\lambda w.\, m)\,\bar{X}_{\vec{m}}
\end{array}
$$

Lemma

The relation $\downarrow_a$ is included in $\simeq$:

$$t \downarrow_a m \;\Rightarrow\; t \simeq m$$

Reduction of unification problems

- part:

$$\frac{P_1 \rightsquigarrow P_2}{P_1, P \rightsquigarrow P_2, P}$$

- bind:

$$\frac{P_1 \rightsquigarrow P_2}{(\nu\omega)\ P_1 \rightsquigarrow (\nu\omega)\ P_2}$$

- eval:

$$t_1 \approx t_2 \rightsquigarrow t_1{\downarrow_a} \approx t_2{\downarrow_a}$$

- l/l:

$$\lambda z^\sigma .\, m_1^\tau \approx \lambda z^\sigma .\, m_2^\tau \;\rightsquigarrow\; (\nu z)\; m_1^\tau \approx m_2^\tau$$

- x/x:

$$x^\tau \approx x^\tau \;\rightsquigarrow\; \varnothing$$

- x/l:

$$x^{\sigma\to\tau} \approx \lambda z^\sigma .\, m^\tau \;\rightsquigarrow\; (\nu z)\; x^{\sigma\to\tau}\, z^\sigma \approx m^\tau$$

- $\neg x/\neg x$:

$$(\lambda w^{\tau'} .\, m_1^\tau)\,(x^{\sigma\to\tau'}\; v_1^\sigma) \approx (\lambda w^{\tau'} .\, m_2^\tau)\,(x^{\sigma\to\tau'}\; v_2^{\sigma 2})$$

$$\rightsquigarrow\; v_1^{\sigma 1} \approx v_2^{\sigma 2},\; (\nu w)\; m_1^\tau \approx m_2^\tau$$

Others cases where unknowns do not appear are impossible:

- $$x_1^{\tau} \approx x_2^{\tau} \quad \text{if } x_1^{\tau} \neq x_2^{\tau}$$

- $$(\lambda w^{\tau_1}.\, m_1^{\tau})\,(x_1^{\sigma_1 \to \tau_1}\; v_1^{\sigma_1}) \approx x_2^{\tau}$$

- $$(\lambda w^{\tau_1}.\, m_1^{\tau})\,(x_1^{\sigma_1 \to \tau_1}\; v_1^{\sigma_1}) \approx I^{\tau}$$

- $$(\lambda w^{\tau_1}.\, m_1^{\tau})\,(x_1^{\sigma_1 \to \tau_1}\; v_1^{\sigma_1}) \approx (\lambda w^{\tau_2}.\, m_2^{\tau})\,(x_2^{\sigma_2 \to \tau_2}\; v_2^{\sigma_2})$$
  $$\text{if } x_1^{\sigma_1 \to \tau_1} \neq x_2^{\sigma_2 \to \tau_2}$$

Some rules to guess what the unknowns should be substituted with:

- if $v^\tau \in \Gamma$

$$(\nu \underline{X})\ P \ \rightsquigarrow\ P[\underline{X}^\tau_\Gamma := v^\tau]$$

- if $z \notin \Gamma$

$$(\nu \underline{X})\ P \ \rightsquigarrow\ (\nu \bar{Y})\ P[\underline{X}^{\sigma\to\tau}_\Gamma := \lambda z^\sigma .\ \bar{Y}^\tau_{z^\sigma,\Gamma}]$$

- if $t^\tau \in \Gamma$

$$(\nu \bar{X})\ P \ \leadsto\ P[\bar{X}_\Gamma^\tau := t^\tau]$$

- if $z \notin \Gamma$

$$(\nu \bar{X})\ P \ \leadsto\ (\nu \bar{Y})\ P[\bar{X}_\Gamma^{\sigma \to \tau} := \lambda z^\sigma .\ \bar{Y}^\tau_{z^\sigma, \Gamma}]$$

- if $w \notin \Gamma$ and $t^{\sigma' \to \tau'} \in \Gamma$

$$(\nu \bar{X})\ P \ \leadsto\ (\nu \bar{Y})\ (\nu \underline{Z})\ P[\underline{X}_\Gamma^{\sigma \to \tau} := (\lambda w^{\tau'} .\ \bar{Y}^\tau_{w^{\tau'}, \Gamma})(t^{\sigma' \to \tau'}\ \underline{Z}^{\sigma'}_{t^\sigma, \Gamma})]$$