Radbound University

Type-and-Scope Safe Programs and their Proofs

Guillaume Allais James Chapman Conor McBride James McKinna

CPP, 2017-01-17

Motivations

- Formally studying PLs
  - Representation of Terms / Typing derivations
  - With good properties: closed under renaming and substitution, normalising
  - Themselves with good properties
- Writing DSLs
  - Strong guarantees (type, scope safety)
  - With ASTs we can inspect (optimise, compile)

Simple Types

Minimal system: A record type, a sum type and function spaces.

```
data Ty : Set where
  `1 `2 : Ty
  _`→_  : Ty → Ty → Ty

data Cx (ty : Set) : Set where
  ε     : Cx ty
  _•_   : Cx ty → ty → Cx ty
```

Typed de Bruijn indices

data Var $(\tau : ty)$: Cx $ty \rightarrow$ Set where

ze : – ∀ Γ. Var τ (Γ • τ)

[$\tau \vdash$ Var τ]

su : – ∀ Γ σ. Var τ Γ → Var τ (Γ • σ)

[Var $\tau \dot{\rightarrow} (\sigma \vdash$ Var $\tau)$]

ASTs type and scope correct by construction

```
data Tm : Ty → Cx Ty → Set where
  `var  : [ Var σ ∸→                 Tm σ         ]
  _`$_  : [ Tm (σ `→ τ) ∸→ Tm σ ∸→   Tm τ         ]
  `λ    : [ σ ⊢ Tm τ ∸→              Tm (σ `→ τ)  ]
```

A Generic Notion of Environment

$\mathsf{record}\ \_\text{–}\mathsf{Env}\ (\Gamma\ :\ \mathsf{Cx}\ ty)\ (\mathscr{V}\ :\ \mathsf{Model})\ (\Delta\ :\ \mathsf{Cx}\ ty)\ :\ \mathsf{Set}$

$\quad \mathsf{where}\ \ \mathsf{constructor}\ \mathsf{pack}$

$\qquad\qquad \mathsf{field}\ \mathsf{lookup}\ :\ \mathsf{Var}\ \sigma\ \Gamma \to \mathscr{V}\ \sigma\ \Delta$

Goguen & McKinna: Conspicuously similar functions

$$
\begin{aligned}
&\mathsf{ren} : (\Gamma\ {-}\mathsf{Env})\ \mathsf{Var}\ \Delta \to \mathsf{Tm}\ \sigma\ \Gamma \to \mathsf{Tm}\ \sigma\ \Delta\\
&\mathsf{ren}\ \rho\ (\text{`}\mathsf{var}\ v) &&= \mathsf{ren}[\![\mathsf{var}]\!]\ (\mathsf{lookup}\ \rho\ v)\\
&\mathsf{ren}\ \rho\ (t\ \text{`}\$\ u) &&= \mathsf{ren}\ \rho\ t\ \text{`}\$\ \mathsf{ren}\ \rho\ u\\
&\mathsf{ren}\ \rho\ (\text{`}\lambda\ t) &&= \text{`}\lambda\ (\mathsf{ren}\ (\mathsf{renextend}\ \rho)\ t)
\end{aligned}
$$

Goguen & McKinna: Conspicuously similar functions

$$
\begin{aligned}
&\mathsf{sub} : (\Gamma \mathrel{-}\mathsf{Env})\ \mathsf{Tm}\ \Delta \to \mathsf{Tm}\ \sigma\ \Gamma \to \mathsf{Tm}\ \sigma\ \Delta\\
&\mathsf{sub}\ \rho\ (\text{`}\mathsf{var}\ v) = \mathsf{sub}[\![\mathsf{var}]\!]\ (\mathsf{lookup}\ \rho\ v)\\
&\mathsf{sub}\ \rho\ (t\ \text{`}\$\ u) = \mathsf{sub}\ \rho\ t\ \ \text{`}\$\ \mathsf{sub}\ \rho\ u\\
&\mathsf{sub}\ \rho\ (\text{`}\lambda\ t) = \text{`}\lambda\ (\mathsf{sub}\ (\mathsf{subextend}\ \rho)\ t)
\end{aligned}
$$

$\square : (\mathsf{Cx}\ ty \to \mathsf{Set}) \to (\mathsf{Cx}\ ty \to \mathsf{Set})$
$(\square\ S)\ \Gamma = \Gamma \subseteq \Delta \to S\ \Delta$

$\mathsf{Thinnable}\ S = \Gamma \subseteq \Delta \to (S\ \Gamma \to S\ \Delta)$

```
record Syntactic (𝒱 : Model) : Set where
  field th     : (σ : Ty) → Thinnable (𝒱 σ)
        var₀   : [ σ ⊢ 𝒱 σ      ]
        ⟦var⟧  : [ 𝒱 σ ∸→ Tm σ ]
```

Implementing the traversal Once and For All

$$
\begin{aligned}
&\mathsf{syn} : (\mathcal{S} : \mathsf{Syntactic}\ \mathscr{V}) \to (\Gamma\ {-}\mathsf{Env})\ \mathscr{V}\ \Delta \to \mathsf{Tm}\ \sigma\ \Gamma \to \mathsf{Tm}\ \sigma\ \Delta \\
&\mathsf{syn}\ \mathcal{S}\ \rho\ ({}^\backprime\mathsf{var}\ v) = \mathsf{Syntactic.}[\![\mathsf{var}]\!]\ \mathcal{S}\ (\mathsf{lookup}\ \rho\ v) \\
&\mathsf{syn}\ \mathcal{S}\ \rho\ (t\ {}^\backprime\$\ u) = \mathsf{syn}\ \mathcal{S}\ \rho\ t\ {}^\backprime\$\ \mathsf{syn}\ \mathcal{S}\ \rho\ u \\
&\mathsf{syn}\ \mathcal{S}\ \rho\ ({}^\backprime\lambda\ t) = {}^\backprime\lambda\ (\mathsf{syn}\ \mathcal{S}\ (\mathsf{synextend}\ \mathcal{S}\ \rho)\ t)
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{synextend} : \forall\ (\mathcal{S} : \mathsf{Syntactic}\ \mathscr{V}) \to \\
&\qquad (\Gamma\ {-}\mathsf{Env})\ \mathscr{V}\ \Delta \to (\Gamma \bullet \sigma\ {-}\mathsf{Env})\ \mathscr{V}\ (\Delta \bullet \sigma) \\
&\mathsf{synextend}\ \mathcal{S}\ \rho = \rho'\ {}^\backprime\!\bullet\ \mathsf{var} \\
&\quad \mathsf{where}\ \mathsf{var} = \mathsf{Syntactic.var_0}\ \mathcal{S} \\
&\qquad\qquad \rho' = \mathsf{pack}\ \$\ \mathsf{Syntactic.th}\ \mathcal{S}\ \_\ (\mathsf{pack\ su}) \circ \mathsf{lookup}\ \rho
\end{aligned}
$$

Is that it? Not quite.

- Other interesting instance?
- Properties of these traversals? (2*n fusion lemmas)
- What if I don't program in Agda?
- Generic boilerplate for all syntaxes with binding?

$$\square : (\mathsf{Cx}\ ty \to \mathsf{Set}) \to (\mathsf{Cx}\ ty \to \mathsf{Set})$$
$$(\square\ S)\ \Gamma = \Gamma \subseteq \Delta \to S\ \Delta$$
$$\mathsf{Kr}\ (\sigma \mathbin{`\!\!\to} \tau) = \square\ (\mathsf{Kr}\ \sigma \mathbin{\dot\to} \mathsf{Kr}\ \tau)$$

$$\mathsf{sem} : (\Gamma\ \mathsf{-Env})\ \mathsf{Kr}\ \Delta \to \mathsf{Tm}\ \sigma\ \Gamma \to \mathsf{Kr}\ \sigma\ \Delta$$
$$\mathsf{sem}\ \rho\ (`\mathsf{var}\ v) = \mathsf{sem}[\![\mathsf{var}]\!]\ (\mathsf{lookup}\ \rho\ v)$$
$$\mathsf{sem}\ \rho\ (t \mathbin{`\$} u) = \mathsf{sem\$}\ t\ u\ (\mathsf{sem}\ \rho\ t)\ (\mathsf{sem}\ \rho\ u)$$
$$\mathsf{sem}\ \rho\ (`\lambda\ t) = \mathsf{sem}\lambda\ t\ (\lambda\ \rho \to \mathsf{sem}\ \rho\ t)\ (\mathsf{semextend}\ \rho)$$

$$
\begin{aligned}
&\mathsf{sub} : (\Gamma\ \mathsf{-Env})\ \mathsf{Tm}\ \Delta \to \mathsf{Tm}\ \sigma\ \Gamma \to \mathsf{Tm}\ \sigma\ \Delta\\
&\mathsf{sub}\ \rho\ (\text{`}\mathsf{var}\ v) \quad = \mathsf{sub}[\![\mathsf{var}]\!]\ (\mathsf{lookup}\ \rho\ v)\\
&\mathsf{sub}\ \rho\ (t\ \text{`}\$\ u) \quad = \mathsf{sub}\ \rho\ t\ \ \text{`}\$\ \mathsf{sub}\ \rho\ u\\
&\mathsf{sub}\ \rho\ (\text{`}\lambda\ t) \qquad = \text{`}\lambda\ (\mathsf{sub}\ (\mathsf{subextend}\ \rho)\ t)
\end{aligned}
$$

An Abstract Notion of Semantics

```
record Semantics (𝒱 𝒞 : `Model) : Set where
  field
```

```
record Semantics (𝒱 𝒞 : `Model) : Set where
  field

  th     : ∀ σ → Thinnable (𝒱 σ)
  ⟦var⟧  : ∀ σ → [ 𝒱 σ ∸→ 𝒞 σ ]
```

```
record Semantics (𝒱 𝒞 : `Model) : Set where
  field

  th     : ∀ σ → Thinnable (𝒱 σ)
  ⟦var⟧  : ∀ σ → [ 𝒱 σ ∸→ 𝒞 σ ]

  ⟦λ⟧    :  [ □ (𝒱 σ ∸→ 𝒞 τ)    ∸→ 𝒞 (σ `→ τ) ]
  _⟦$⟧_  :  [ 𝒞 (σ `→ τ) ∸→ 𝒞 σ  ∸→ 𝒞 τ        ]
```

And a Fundamental Lemma

$$\_\text{–Comp} : \mathsf{Cx}\ \mathsf{Ty} \rightarrow (\mathscr{C} : \mathsf{Model}) \rightarrow \mathsf{Cx}\ \mathsf{Ty} \rightarrow \mathsf{Set}$$
$$(\Gamma\ \text{–Comp})\ \mathscr{C}\ \Delta = \mathsf{Tm}\ \sigma\ \Gamma \rightarrow \mathscr{C}\ \sigma\ \Delta$$

```
- ∀ Γ Δ. (Γ -Env) V Δ → ∀ σ. Tm σ Γ → C σ Δ
```

$$\mathsf{sem} : [\ (\Gamma\ \text{–Env})\ \mathscr{V}\ \dot{\rightarrow}\ (\Gamma\ \text{–Comp})\ \mathscr{C}\ ]$$
$$\mathsf{sem}\ \rho\ (\text{‘var}\ v) = [\![\mathsf{var}]\!]\ (\mathsf{lookup}\ \rho\ v)$$
$$\mathsf{sem}\ \rho\ (t\ \text{‘\$}\ u) = \mathsf{sem}\ \rho\ t\ [\![\$]\!]\ \mathsf{sem}\ \rho\ u$$
$$\mathsf{sem}\ \rho\ (\text{‘}\lambda\ b) = [\![\lambda]\!]\ (\lambda\ \sigma\ v \rightarrow \mathsf{sem}\ (\mathsf{semextend}\ \rho\ \sigma\ v)\ b)$$

$$\mathsf{semextend} : (\Gamma\ \text{–Env})\ \mathscr{V}\ \Delta \rightarrow \Delta \subseteq \Theta \rightarrow \mathscr{V}\ \sigma\ \Theta \rightarrow (\Gamma \bullet \sigma\ \text{–Env})\ \mathscr{V}\ \Theta$$
$$\mathsf{semextend}\ \rho\ \sigma\ v = \mathsf{th[\ th\ ]}\ \sigma\ \rho\ \text{‘}\bullet\ v$$

Various Instances

$$\begin{aligned} \mathsf{Renaming} &: \mathsf{Semantics}\ \mathsf{Var}\ \mathsf{Tm} \\ \mathsf{Substitution} &: \mathsf{Semantics}\ \mathsf{Tm}\ \mathsf{Tm} \end{aligned}$$

Various Instances

Renaming : Semantics Var Tm
Substitution : Semantics Tm Tm

Normalise : Semantics Kr Kr

Various Instances

Renaming : Semantics Var Tm

Substitution : Semantics Tm Tm

Normalise : Semantics Kr Kr

CPS_N : Semantics Var_N Ml_N

Various Instances

$$
\begin{aligned}
\text{Renaming} &: \text{Semantics Var Tm}\\
\text{Substitution} &: \text{Semantics Tm Tm}
\end{aligned}
$$

$$\text{Normalise} : \text{Semantics Kr Kr}$$

$$\text{CPS}_N : \text{Semantics Var}_N\ \text{Ml}_N$$

$$\text{Printing} : \text{Semantics Name Printer}$$

Is that it? Not quite.

- Other interesting instance?
- Properties of these traversals? (2*n fusion lemmas)
- What if I don't program in Agda?
- Generic boilerplate for all syntaxes with binding?

A Relational Interpretation

$\mathscr{R} : \mathsf{Tm}\ \sigma\ \Gamma \to (\Gamma\ \mathsf{-Env})\ \mathscr{V}_{\mathsf{A}}\ \Delta \to (\Gamma\ \mathsf{-Env})\ \mathscr{V}_{\mathsf{B}}\ \Delta \to \mathsf{Set}$

$\mathscr{R}\ t\ \rho_{\mathsf{A}}\ \rho_{\mathsf{B}} = \mathsf{rmodel}\ \mathscr{C}_{\mathsf{R}}\ (\mathsf{sem}_{\mathsf{A}}\ \rho_{\mathsf{A}}\ t)\ (\mathsf{sem}_{\mathsf{B}}\ \rho_{\mathsf{B}}\ t)$

$\mathsf{record}\ \mathsf{Simulation}$

$\quad (\mathscr{S}_{\mathsf{A}} : \mathsf{Semantics}\ \mathscr{V}_{\mathsf{A}}\ \mathscr{C}_{\mathsf{A}})\ (\mathscr{S}_{\mathsf{B}} : \mathsf{Semantics}\ \mathscr{V}_{\mathsf{B}}\ \mathscr{C}_{\mathsf{B}})$

$\quad (\mathscr{V}_{\mathsf{R}} : \text{`}\mathsf{RModel}\ \mathscr{V}_{\mathsf{A}}\ \mathscr{V}_{\mathsf{B}})\ (\mathscr{C}_{\mathsf{R}} : \text{`}\mathsf{RModel}\ \mathscr{C}_{\mathsf{A}}\ \mathscr{C}_{\mathsf{B}}) : \mathsf{Set}\ \mathsf{where}$

A Relational Interpretation

$\mathscr{R} : \mathsf{Tm}\ \sigma\ \Gamma \to (\Gamma\ {-}\mathsf{Env})\ \mathscr{V}_{\mathsf{A}}\ \Delta \to (\Gamma\ {-}\mathsf{Env})\ \mathscr{V}_{\mathsf{B}}\ \Delta \to \mathsf{Set}$

$\mathscr{R}\ t\ \rho_{\mathsf{A}}\ \rho_{\mathsf{B}} = \mathsf{rmodel}\ \mathscr{C}_{\mathsf{R}}\ (\mathsf{sem}_{\mathsf{A}}\ \rho_{\mathsf{A}}\ t)\ (\mathsf{sem}_{\mathsf{B}}\ \rho_{\mathsf{B}}\ t)$

record Simulation

$(\mathscr{S}_{\mathsf{A}} : \mathsf{Semantics}\ \mathscr{V}_{\mathsf{A}}\ \mathscr{C}_{\mathsf{A}})\ (\mathscr{S}_{\mathsf{B}} : \mathsf{Semantics}\ \mathscr{V}_{\mathsf{B}}\ \mathscr{C}_{\mathsf{B}})$

$(\mathscr{V}_{\mathsf{R}} : \text{`}\mathsf{RModel}\ \mathscr{V}_{\mathsf{A}}\ \mathscr{V}_{\mathsf{B}})\ (\mathscr{C}_{\mathsf{R}} : \text{`}\mathsf{RModel}\ \mathscr{C}_{\mathsf{A}}\ \mathscr{C}_{\mathsf{B}}) : \mathsf{Set}$ where

$\mathscr{V}_{\mathsf{R}_{\mathsf{th}}} : \text{`}\forall[\ \mathscr{V}_{\mathsf{R}}\ ]\ \rho_{\mathsf{A}}\ \rho_{\mathsf{B}} \to \text{`}\forall[\ \mathscr{V}_{\mathsf{R}}\ ]\ (\mathsf{th}[\ \mathscr{S}_{\mathsf{A}}.\mathsf{th}\ ]\ \mathit{inc}\ \rho_{\mathsf{A}})\ (\mathsf{th}[\ \mathscr{S}_{\mathsf{B}}.\mathsf{th}\ ]\ \mathit{inc}\ \rho_{\mathsf{B}})$

A Relational Interpretation

$\mathcal{R} : \mathsf{Tm}\ \sigma\ \Gamma \to (\Gamma\ \text{–Env})\ \mathcal{V}_A\ \Delta \to (\Gamma\ \text{–Env})\ \mathcal{V}_B\ \Delta \to \mathsf{Set}$

$\mathcal{R}\ t\ \rho_A\ \rho_B = \mathsf{rmodel}\ \mathcal{C}_R\ (\mathsf{sem}_A\ \rho_A\ t)\ (\mathsf{sem}_B\ \rho_B\ t)$

$\mathsf{record\ Simulation}$

$(\mathcal{S}_A : \mathsf{Semantics}\ \mathcal{V}_A\ \mathcal{C}_A)\ (\mathcal{S}_B : \mathsf{Semantics}\ \mathcal{V}_B\ \mathcal{C}_B)$

$(\mathcal{V}_R\ : \mathsf{`RModel}\ \mathcal{V}_A\ \mathcal{V}_B)\ (\mathcal{C}_R\ : \mathsf{`RModel}\ \mathcal{C}_A\ \mathcal{C}_B) : \mathsf{Set\ where}$

$\mathcal{V}_{R_{th}}\ :\ \mathsf{`\forall[}\ \mathcal{V}_R\ \mathsf{]}\ \rho_A\ \rho_B \to \mathsf{`\forall[}\ \mathcal{V}_R\ \mathsf{]}\ (\mathsf{th[}\ \mathcal{S}_A.\mathsf{th}\ \mathsf{]}\ \mathit{inc}\ \rho_A)\ (\mathsf{th[}\ \mathcal{S}_B.\mathsf{th}\ \mathsf{]}\ \mathit{inc}\ \rho_B)$

$\mathsf{R}[\![\mathsf{var}]\!]\ :\ \mathsf{`\forall[}\ \mathcal{V}_R\ \mathsf{]}\ \rho_A\ \rho_B \to \mathcal{R}\ (\mathsf{`var}\ v)\ \rho_A\ \rho_B$

A Relational Interpretation

$\mathcal{R} : \mathsf{Tm}\ \sigma\ \Gamma \to (\Gamma - \mathsf{Env})\ \mathcal{V}_{A}\ \Delta \to (\Gamma - \mathsf{Env})\ \mathcal{V}_{B}\ \Delta \to \mathsf{Set}$

$\mathcal{R}\ t\ \rho_{A}\ \rho_{B} = \mathsf{rmodel}\ \mathcal{C}_{R}\ (\mathsf{sem}_{A}\ \rho_{A}\ t)\ (\mathsf{sem}_{B}\ \rho_{B}\ t)$

$\mathsf{record\ Simulation}$

$\quad (\mathcal{S}_{A} : \mathsf{Semantics}\ \mathcal{V}_{A}\ \mathcal{C}_{A})\ (\mathcal{S}_{B} : \mathsf{Semantics}\ \mathcal{V}_{B}\ \mathcal{C}_{B})$

$\quad (\mathcal{V}_{R} : \text{`}\mathsf{RModel}\ \mathcal{V}_{A}\ \mathcal{V}_{B})\ (\mathcal{C}_{R} : \text{`}\mathsf{RModel}\ \mathcal{C}_{A}\ \mathcal{C}_{B}) : \mathsf{Set\ where}$

$\quad \mathcal{V}_{R_{th}} : \text{`}\forall[\ \mathcal{V}_{R}\ ]\ \rho_{A}\ \rho_{B} \to \text{`}\forall[\ \mathcal{V}_{R}\ ]\ (\mathsf{th}[\ \mathcal{S}_{A}.\mathsf{th}\ ]\ inc\ \rho_{A})\ (\mathsf{th}[\ \mathcal{S}_{B}.\mathsf{th}\ ]\ inc\ \rho_{B})$

$\quad \mathsf{R}[\![\mathsf{var}]\!] : \text{`}\forall[\ \mathcal{V}_{R}\ ]\ \rho_{A}\ \rho_{B} \to \mathcal{R}\ (\text{`}\mathsf{var}\ v)\ \rho_{A}\ \rho_{B}$

$\quad \mathsf{R}[\![\lambda]\!] : \forall\ (b : \mathsf{Tm}\ \tau\ (\Gamma \bullet \sigma)) \to$

$\qquad (b_{R} : (pr : \Delta \subseteq \Theta) \to \mathsf{rmodel}\ \mathcal{V}_{R}\ u_{A}\ u_{B} \to$

$\qquad\quad \mathcal{R}\ b\ (\mathsf{semextend}\ \mathcal{S}_{A}\ \rho_{A}\ pr\ u_{A})\ (\mathsf{semextend}\ \mathcal{S}_{B}\ \rho_{B}\ pr\ u_{B})) \to$

$\qquad \text{`}\forall[\ \mathcal{V}_{R}\ ]\ \rho_{A}\ \rho_{B} \to \mathcal{R}\ (\text{`}\lambda\ b)\ \rho_{A}\ \rho_{B}$

And a Fundamental Lemma

$$
\begin{aligned}
&\mathsf{sim} : (t : \mathsf{Tm}\ \sigma\ \Gamma) \to \text{`}\forall[\ \mathscr{V}_{\mathsf{R}}\ ]\ \rho_{\mathsf{A}}\ \rho_{\mathsf{B}} \to \\
&\qquad \mathsf{rmodel}\ \mathscr{C}_{\mathsf{R}}\ (\mathsf{sem}_{\mathsf{A}}\ \rho_{\mathsf{A}}\ t)\ (\mathsf{sem}_{\mathsf{B}}\ \rho_{\mathsf{B}}\ t) \\
&\mathsf{sim}\ (\text{`}\mathsf{var}\ v)\quad \rho_{\mathsf{R}} = \mathsf{R}[\![\mathsf{var}]\!]\ v\ \rho_{\mathsf{R}} \\
&\mathsf{sim}\ (f\ \text{`}\$\ t)\quad \rho_{\mathsf{R}} = \mathsf{R}[\![\$]\!]\ (\mathsf{sim}\ f\ \rho_{\mathsf{R}})\ (\mathsf{sim}\ t\ \rho_{\mathsf{R}})\ \rho_{\mathsf{R}} \\
&\mathsf{sim}\ (\text{`}\lambda\ t)\quad \rho_{\mathsf{R}} = \mathsf{R}[\![\lambda]\!]\ (\lambda\ inc\ u_{\mathsf{R}} \to \mathsf{sim}\ t\ (\mathscr{V}_{\mathsf{R}_{\mathsf{th}}}\ inc\ \rho_{\mathsf{R}} \bullet_{\mathsf{R}} u_{\mathsf{R}}))\ \rho_{\mathsf{R}}
\end{aligned}
$$

An interesting corollary

$\mathsf{SimulationNormalise} : \mathsf{Simulation}\ \mathsf{Normalise}\ \mathsf{Normalise}\ \mathsf{PER'}\ \mathsf{PER'}$

$\mathsf{refl}_{\mathsf{Kr}} : \forall\ (t : \mathsf{Tm}\ \sigma\ \Gamma)\ (\rho : (\Gamma\ \mathsf{-Env})\ \mathsf{Kr}\ \Delta) \to$
$\quad \mathsf{let}\ T = \mathsf{Eval.sem}\ \mathsf{Normalise}\ \rho\ t\ \mathsf{in}$
$\quad \text{`}\forall[\ \mathsf{PER'}\ ]\ \rho\ \rho \to \mathsf{PER}\ \sigma\ T\ T$
$\mathsf{refl}_{\mathsf{Kr}}\ t\ \rho\ \rho_{\mathsf{R}} = \mathsf{Simulate.sim}\ \mathsf{SimulationNormalise}\ t\ \rho_{\mathsf{R}}$

A Generic Fusion Theorem

$$\mathsf{th}_{\mathsf{Tm}}\ \sigma\ \rho'\ (\mathsf{th}_{\mathsf{Tm}}\ \sigma\ \rho\ t) \equiv \mathsf{th}_{\mathsf{Tm}}\ \sigma\ (\rho'\ [\circ]\ \rho)\ t$$
$$\mathsf{th}_{\mathsf{Tm}}\ \sigma\ \rho'\ (\mathsf{subst}\ \rho\ t) \equiv \mathsf{subst}\ (\mathsf{map}_{\mathsf{Env}}\ (\mathsf{th}_{\mathsf{Tm}}\ \_\ \rho')\ \rho)\ t$$
$$\mathsf{subst}\ \rho'\ (\mathsf{th}_{\mathsf{Tm}}\ \sigma\ \rho\ t) \equiv \mathsf{subst}\ (\rho'\ [\circ]\ \rho)\ t$$
$$\mathsf{subst}\ \rho'\ (\mathsf{subst}\ \rho\ t) \equiv \mathsf{subst}\ (\mathsf{map}_{\mathsf{Env}}\ (\mathsf{subst}\ \rho')\ \rho)\ t$$

$$\mathsf{PER}\ \sigma\ (\mathsf{nbe}\ \rho'\ (\mathsf{subst}\ \rho\ t))\ (\mathsf{nbe}\ (\mathsf{map}_{\mathsf{Env}}\ (\mathsf{nbe}\ \rho')\ \rho)\ t)$$

Is that it? Not quite.

- Other interesting instance?
- Properties of these traversals? (2*n fusion lemmas)
- What if I don't program in Agda?
- Generic boilerplate for all syntaxes with binding?

Using this somewhere else

The programming part of this talk can be implemented in Haskell:

https://github.com/gallais/type-scope-semantics/

Is that it? Not quite.

- Other interesting instance?
- Properties of these traversals? (2*n fusion lemmas)
- What if I don't program in Agda?
- Generic boilerplate for all syntaxes with binding?