Type-and-Scope Safe Programs and Their Proofs

Guillaume Allais
gallais@cs.ru.nl
Radboud University,
The Netherlands

James Chapman Conor McBride
{james.chapman,conor.mcbride}@strath.ac.uk
University of Strathclyde, UK

James McKinna
james.mckinna@ed.ac.uk
University of Edinburgh, UK

Abstract

We abstract the common type-and-scope safe structure from computations on λ-terms that deliver, e.g., renaming, substitution, evaluation, CPS-transformation, and printing with a name supply. By exposing this structure, we can prove generic simulation and fusion lemmas relating operations built this way. This work has been fully formalised in Agda.

Categories and Subject Descriptors D.2.4 [*Software / Program Verification*]: Correctness Proofs; D.3.2 [*Language Classifications*]: Applicative (functional) languages; F.3.2 [*Semantics of Programming Languages*]: Denotational semantics, Partial evaluation

```
ren : (∀σ. Var σ Γ → Var σ Δ) → (∀σ. Tm σ Γ → Tm σ Δ)
ren ρ (`var v)  = `var (ρ v)
ren ρ (f `$ t)  = ren ρ f `$ ren ρ t
ren ρ (`λ b)    = `λ (ren ((su ∘ ρ) −, ze) b)

sub : (∀σ. Var σ Γ → Tm σ Δ) → (∀σ. Tm σ Γ → Tm σ Δ)
sub ρ (`var v)  = ρ v
sub ρ (f `$ t)  = sub ρ f `$ sub ρ t
sub ρ (`λ b)    = `λ (sub ((ren su ∘ ρ) −, `var ze) b)
```

Figure 1. Renaming and Substitution for the STλC