Properties of λ2

Definition: α-conversion, α-equivalence

(1a) (Renaming of term variable)

$\lambda x:\sigma . M =_\alpha \lambda y:\sigma . M^{x\to y}$ if $y \notin FV(M)$ and $y \notin Bi(M)$

(1b) (Renaming of type variable)

$\lambda \alpha:* . M =_\alpha \lambda \beta:* . M[\alpha:=\beta]$ if β does not occur in M

$\Pi \alpha:* . M =_\alpha \Pi \beta:* . M[\alpha:=\beta]$ if β does not occur in M

(2), (3a), (3b), (3c) (Compatibility, Reflexivity, Symmetry, Transitivity)
As in λ→

Definition: β-reduction, $\to_\beta$, $\twoheadrightarrow_\beta$, $=_\beta$

(1a) (Basis, first order)

$(\lambda x:\sigma . M)N \to_\beta M[x:=N]$

(1b) (Basis, second order)

$(\lambda \alpha:* . M)T \to_\beta M[\alpha:=T]$

(2) (Compatibility)
As in λ→

Example: $(\lambda \alpha:* . \lambda f:\alpha\to\alpha . \lambda x:\alpha . f(f x))$ nat suc two $\to_\beta$

$(\lambda f: nat \to nat . \lambda x: nat . f(f x))$ suc two $\to_\beta$

$(\lambda x: nat . suc(suc\ x))$ two $\to_\beta$

suc (suc two)

All four terms have type nat, β-reduction does not change the type which can be seen in two ways:

(1) Give type derivations for all terms

(2) Subject reduction lemma (still valid in λ2)

The following lemmas still hold in λ2:

- Free variable lemma (f.v. are declared in context)
- Thinning lemma (context may be expanded artificially)
- Condensing lemma (context may be shrunk to only f.v.)
- Generation lemma (derivation rules are only ways to construct judgements)
- Subterm lemma (subterms of legal terms are legal)
- Uniqueness of types lemma (if a term has two types, types coincide)
- Substitution lemma (substituting variable with term of same type doesn't change type)
- Church-Rosser theorem (confluence, $M \twoheadrightarrow_\beta N_1 \twoheadrightarrow_\beta N$, $M \twoheadrightarrow_\beta N_2 \twoheadrightarrow_\beta N$)
- Subject reduction lemma (β-reduction doesn't change type)
- Strong normalization theorem (every legal term is strongly ~~norm~~ normalizing)
- Permutation lemma, if the permuted context is still valid! (declarations in context may be permuted)

Type inference (implicit typing) is no longer decidable in λ2!