Equalities on Lambda Terms

Peter Thieman

03.07.2015

There are different notions of equality on untyped lambda terms. Warning: there are no standard symbols. You need to check the context to find out which equality is meant.

- Syntactic equality $M = N$
  Uninteresting because it differentiates too many terms which behave the same, like $\lambda x.x \neq \lambda y.y$
- Alpha equivalence (i.e., equal up to renaming of bound variables) $M =_\alpha N$
  Alpha equivalence is included in all interesting notions of equality on lambda terms.
- Definitional equality $M =_\beta N$
  Zero or more alpha and beta reductions may be applied anywhere in the term and in any direction to transform M into N.
- Extensional definitional equality $M =_{\beta\eta} N$
  In addition to beta reduction, eta reduction η may be applied anywhere in the term:
  $$u \longrightarrow \lambda x.ux \quad \text{if} \quad x \notin \mathsf{free}(u)$$
  Eta reduction requires extensional models, that is, models in which $\forall x.f(x) = g(x)$ implies $f = g$.
- Contextual equivalence or observational equivalence
  $M \equiv N$ iff, for all contexts C, no difference can be observed between $C[M]$ and $C[N]$.
  This notion is parametric over the notion of observation, which can be as weak as observing termination (i.e., $C[M] \downarrow$ iff $C[N] \downarrow$). For contexts of type number, the definition could ask for the same number to be returned upon termination.