Normalization

(Weak) Normalization

Normalization = there is a terminating sequence of reductions.

We have seen that in that outermost is normalizing if any is.

The typed lambda calculus is normalizing.

(We have already seen that it is terminating, so this is a weaker result.)

Difficulty

- Terms grow
  - $(\lambda f.\lambda x.f(fx))B \Rightarrow \lambda x.B(Bx)$

- Redexes born
  - $(\lambda f.\lambda x.f(fx))(\lambda y.M) \Rightarrow \lambda x.(\lambda y.M)((\lambda y.M)x)$

- Redexes multiply
  - $(\lambda f.\lambda x.f(fx))((\lambda y.M)B) \Rightarrow \lambda x.((\lambda y.M)B)(((\lambda y.M)B)x)$

Preservation

- Contraction preserves type
- Reduction preserves type

Redex Creation

- Born
  - $(\lambda x.C[xN]))(\lambda y.M) ➨ C'[(\lambda y.M)N']$
- Contracted
  - $(\lambda x.\lambda y.M[x])NP ➨ (\lambda y.M[N])P$
- Extracted
  - $(\lambda x.x)(\lambda y.M)N ➨ (\lambda y.M)N$

Nesting Height

Base type: h(o) = 0

Arrow type: $h(\sigma \to \tau) = \max(h(\sigma)+1, h(\tau))$

Redex Reduction

- Born
  - (λx.C[xN]))(λy.M) ➨ C'[(λy.M)N']

    $(\sigma \rightarrow \tau) \rightarrow \theta$ $\sigma \rightarrow \tau$
- Contracted
  - (λx.λy.M[x])NP ➨ (λy.M[N])P

    $\theta \rightarrow (\sigma \rightarrow \tau)$ $\sigma \rightarrow \tau$
- Extracted
  - (λx.x)(λy.M)N ➨ (λy.M)N

    $(\sigma \rightarrow \tau) \rightarrow (\sigma \rightarrow \tau)$ $\sigma \rightarrow \tau$

Normalization Strategy (Turing)

Choose an innermost (rightmost) maximal (highest function type) redex.

Multiset Order

Proof

Look at multiset of redex heights.

No new maximal redexes created by a step.

One maximal redex removed.
Many smaller may be created.

Redex Creation

- Born
  - $(\lambda x.\ ..xB..))(\lambda y.M) \Rightarrow ..(\lambda y.M)B..$

- Duplicated
  - $(\lambda x.\ ..x..x..)((\lambda y.M)B) \Rightarrow ..(\lambda y.M)B..(\lambda y.M)B..$

- Extracted
  - $(\lambda x.x)(\lambda y.M)B \Rightarrow (\lambda y.M)B$
  - $(\lambda x.(\lambda y.M))AB \Rightarrow (\lambda y.M)B$

Richer Types

- Union (sum) types
- Intersection (product) types
- Dependent types
- Polymorphic types
- Types of types

Lambda Cube

extensions