# Chasing Bottoms

Nils Anders Danielsson Patrik Jansson

Chalmers

# Context

- Cover project.
- Verification of real Haskell programs.
- Haskell non-strict $\Rightarrow$ partial and infinite values relatively common.

# Proof methods

- Fixpoint induction.
- The approximation lemma.
- Coinduction.
- Fusion.

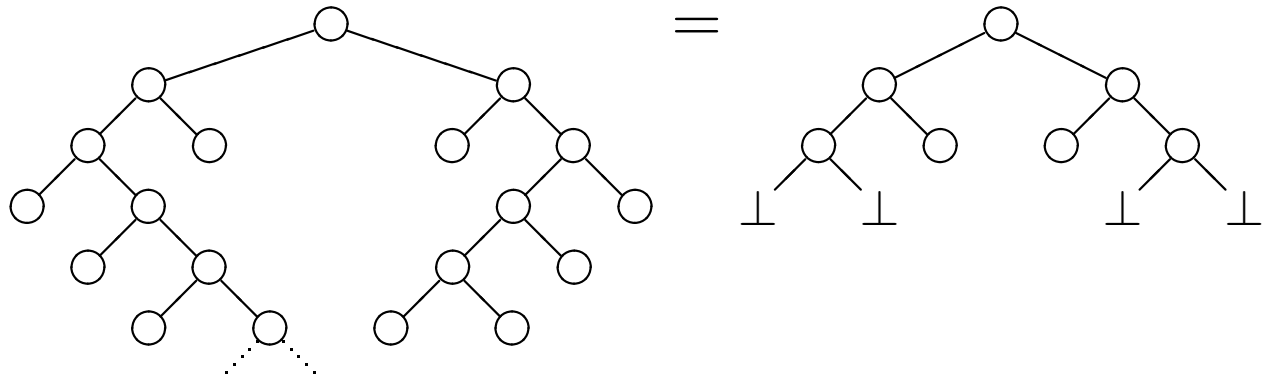
# Caveats

- $\eta$-equality is not valid.
  - $\bot \neq \lambda x.\bot$
- No surjective pairing.
  - $(fst\ \bot, snd\ \bot) = (\bot, \bot) \neq \bot$
- Strange pattern matching semantics.
  - $\lambda True\ x.x \neq \lambda True.\lambda x.x$
- Typical $Monad$ instances not monads.
  - $\bot \gg\!= return = \lambda x.\bot \neq \bot$

# Testing partial values

- $isBottom :: a \to Bool$
- Several flavours of bottom:
  - Non-termination.
  - $error$ "..."
  - Pattern match failure.
- $(isBottom :: a \to IO\ Bool)$
- $\hat{=}, \hat{\sqsubseteq} :: Data\ a \Rightarrow a \to a \to Bool$

# Testing infinite values

- $approx :: Nat \to T \to T$
- $approx\ 3$ 

- The approximation lemma:
  - $t_1 = t_2$ iff $\forall n :: Nat.\ approx\ n\ t_1 = approx\ n\ t_2$

# Conclusions

- Can test and prove, but tricky.
- Approximate semantics might be nice.
  - $\bot \approx \lambda x.\bot$
- For more details, see MPC2004 paper.