CMSC22100 HW #6

Jaime Arana-Rochel

May 23, 2014

Problem 2: Orthogonal Language Features all Stuck Together

Theorem [Type Soundness]. *Well typed terms in the simply-typed lambda calculus do not get stuck. That is, a term $(t : T)$ can alway take a single-step evaluation until it either reaches a value, an error, or it diverges.*

Theorem [Preservation]. *Given:*

$$\Gamma \,|\, \Sigma \vdash t : T$$

$$\Gamma \,|\, \Sigma \vdash \mu$$

$$t \,|\, u \longrightarrow t' \,|\, u'$$

then for some $\Sigma' \supseteq \Sigma$

$$\Gamma \,|\, \Sigma' \vdash t' : T$$

$$\Gamma \,|\, \Sigma' \vdash \mu'$$

Theorem [Progress]. *Given:*

$$\Gamma \,|\, \Sigma \vdash t : T$$

then either t is a value, an error, or for $(\phi \mid \Sigma \vdash \mu)$ then $t \,|\, u \longrightarrow t' \,|\, u'$

Problem 3: Record Subtyping

For the case of double covariance:
$(\lambda r : \{x : Nat, y : Bool\}.\{x = r.x + r.x,\ y = r.y\})\ (\{y = true\})$
This is well typed according to double covariance, but it's obvious from looking at it that this is impossible to evaluate because the function requires a record of type {x:Nat,y:Bool}, but only receives a record of type {y:Bool}.

For the case of double contra-variance:
$(\lambda r : \{x : Nat\}.\{x = r.x + r.x,\ y = r.y\})\ (\{x = 1, y = 2\})$
Again, this is well typed and the arguments check out. However in this case, it's the output type that will not allow this function to be evaluated.