Diagonal

kagan

May 8, 2023

Contents

theory *Cantor* **imports** *Main HOL.Fun*
begin

theorem *Abstracted-Cantor*:
  fixes $f :: 'b \Rightarrow 'a \Rightarrow 'c$ **and** $\alpha :: 'c \Rightarrow 'c$ **and** $\beta :: 'a \Rightarrow 'b$ **and** $\beta\text{-}c :: 'b \Rightarrow 'a$
  assumes *surjectivity*: *surj f*
  and *no-fixed-point*: $\forall\, y.\ \alpha\ y \neq y$
  and *right-inverse*: $\forall\, s.\ \beta\ (\beta\text{-}c\ s) = s$
  shows *False*
proof −
  from *surjectivity* **have** $\forall\, h :: 'a \Rightarrow 'c.\ \exists\, t.\ h = f\ t$ **by** *auto*
  hence $\exists\, t.\ (\alpha \circ (\lambda t'.\ f\ (\beta\ t')\ t')) = f\ t$ **by** *simp*
  then obtain *t0* **where** $(\alpha \circ (\lambda t'.\ f\ (\beta\ t')\ t')) = f\ t0$ **..**
  hence $(\alpha \circ (\lambda t'.\ f\ (\beta\ t')\ t'))\ (\beta\text{-}c\ t0) = f\ t0\ (\beta\text{-}c\ t0)$ **by** (*rule arg-cong*)

  hence $\alpha\ (f\ t0\ (\beta\text{-}c\ t0)) = f\ t0\ (\beta\text{-}c\ t0)$ **using** *right-inverse* **by** *simp*
  thus *False* **using** *no-fixed-point* **by** *simp*
qed

theorem *Generalized-Cantor*:
  fixes $alpha :: 'b \Rightarrow 'b$ **and** $f :: 'a \Rightarrow 'a \Rightarrow 'b$
  assumes *surjectivity*: *surj f*
  and *no-fixed-point*: $\forall\, y.\ alpha\ y \neq y$
  shows *False*

  apply(*rule Abstracted-Cantor*[*of f alpha* $\lambda x.\ x\ \lambda x.\ x$])
  apply(*auto simp add*: *no-fixed-point surjectivity*)
  done

fun $not :: bool \Rightarrow bool$ **where**
not True = *False* |

not False = *True*

theorem *Classic-Cantor*:
  fixes *f* :: *'a* ⇒ *'a* ⇒ *bool*
  assumes *surjectivity*: *surj f*
  shows *False*
  apply(*rule Generalized-Cantor*[*of f not*])
  apply(*auto simp add*: *surjectivity*)
  done

theorem *Classic-Nat-Cantor*:
  fixes *f* :: *nat* ⇒ *nat* ⇒ *bool*
  assumes *surjectivity*: *surj f*
  shows *False*
  apply(*rule Classic-Cantor*[*of f*])
  apply(*simp add*: *surjectivity*)
  done

end