```
data Value : ∀{Γ A} → Γ ⊢ A → Set where
  V-Lambda : ∀{Γ A B} {F : Γ , A ⊢ B} → Value (Lambda {Γ} F)
  V-Lit : ∀{Γ} {i : ℤ} → Value (Lit {Γ} i)
  V-Skip : ∀{Γ} → Value (Skip {Γ})

-- Renaming
ext : ∀{Γ Δ} → (∀{A} → A ∈ Γ → A ∈ Δ)
             → (∀{A B} → B ∈ Γ , A → B ∈ Δ , A)
ext ρ Zero = Zero
ext ρ (Suc x) = Suc (ρ x)

rename : ∀{Γ Δ} → (∀{A} → A ∈ Γ → A ∈ Δ)
                → (∀{A} → Γ ⊢ A → Δ ⊢ A)
rename ρ (Var a) = Var (ρ a)
rename ρ (Lambda f) = Lambda (rename (ext ρ) f)
rename ρ (Sub a A≤:B) = Sub (rename ρ a) A≤:B
rename ρ (App f e) = App (rename ρ f) (rename ρ e)
rename ρ Skip = Skip
rename ρ (Seq c₁ c₂) = Seq (rename ρ c₁) (rename ρ c₂)
rename ρ (NewVar c) = NewVar (rename (ext ρ) c)
rename ρ (Assign a e) = Assign (rename ρ a) (rename ρ e)
rename ρ (Lit i) = Lit i
rename ρ (Neg e) = Neg (rename ρ e)
rename ρ (Plus e₁ e₂) = Plus (rename ρ e₁) (rename ρ e₂)
```