```
-- Simultaneous substitution
exts : ∀{Γ Δ} → (∀{A} → A ∈ Γ → Δ ⊢ A)
               → (∀{A B} → B ∈ Γ , A → Δ , A ⊢ B)
exts σ Zero = Var Zero
exts σ (Suc x) = rename Suc (σ x)

subst : ∀{Γ Δ} → (∀{A} → A ∈ Γ → Δ ⊢ A)
               → (∀{A} → Γ ⊢ A → Δ ⊢ A)
subst σ (Var A∈Γ) = σ A∈Γ
subst σ (Sub Γ⊢A A≤:B) = Sub (subst σ Γ⊢A) A≤:B
subst σ (Lambda Γ,A⊢B) = Lambda (subst (exts σ) Γ,A⊢B)
subst σ (App Γ⊢A Γ⊢B) = App (subst σ Γ⊢A) (subst σ Γ⊢B)
subst σ Skip = Skip
subst σ (Seq Γ⊢c₁ Γ⊢c₂) = Seq (subst σ Γ⊢c₁) (subst σ Γ⊢c₂)
subst σ (NewVar Γ⊢c) = NewVar (subst (exts σ) Γ⊢c)
subst σ (Assign Γ⊢i Γ⊢e) = Assign (subst σ Γ⊢i) (subst σ Γ⊢e)
subst σ (Lit Γ⊢i) = Lit Γ⊢i
subst σ (Neg Γ⊢i) = Neg (subst σ Γ⊢i)
subst σ (Plus Γ⊢i₁ Γ⊢i₂) = Plus (subst σ Γ⊢i₁) (subst σ Γ⊢i₂)

-- Single substitution
_[_] : ∀{Γ A B} → Γ , B ⊢ A → Γ ⊢ B → Γ ⊢ A
_[_] {Γ} {A} {B} N M = subst {Γ , B} {Γ} σ {A} N
  where
  σ : ∀ {A} → A ∈ Γ , B → Γ ⊢ A
  σ Zero = M
  σ (Suc x) = Var x

-- Reduction
data _⟶_ : ∀{Γ A} → (Γ ⊢ A) → (Γ ⊢ A) → Set where
  App-cong₁ : ∀{Γ A B} {F F' : Γ ⊢ A ⇒ B} {E : Γ ⊢ A}
                       → F ⟶ F' → App F E ⟶ App F' E
  App-cong₂ : ∀{Γ A B} {V : Γ ⊢ A ⇒ B} {E E' : Γ ⊢ A}
                       → Value V → E ⟶ E' → App V E ⟶ App V E'
  Lambda-β : ∀{Γ A B} {F : Γ , A ⊢ B} {V : Γ ⊢ A}
                       → Value V → App (Lambda F) V ⟶ F [ V ]
```