```
fmap-⇒ : ∀ {P Q sd sd'} → (P ⇒s Q) sd → sd ≤s sd' → (P ⇒s Q) sd'
fmap-⇒ P⇒Q sd≤ssd' sd'≤ssd'' p = P⇒Q (≤s-trans sd≤ssd' sd'≤ssd'') p

fmap-ty : ∀ {A sd sd'} → ⟦ A ⟧ty sd → sd ≤s sd' → ⟦ A ⟧ty sd'
fmap-ty {comm} = fmap-⇒ {Compl} {Compl}
fmap-ty {intexp} = fmap-⇒ {Intcompl} {Compl}
fmap-ty {intacc} = fmap-⇒ {Compl} {Intcompl}
fmap-ty {intvar} ( exp , acc ) sd≤ssd' =
  ( fmap-ty {intexp} exp sd≤ssd' , fmap-ty {intacc} acc sd≤ssd' )
fmap-ty {A ⇒ B} = fmap-⇒ {⟦ A ⟧ty} {⟦ B ⟧ty}

fmap-ctx : ∀ {Γ sd sd'} → ⟦ Γ ⟧ctx sd → sd ≤s sd' → ⟦ Γ ⟧ctx sd'
fmap-ctx { · } unit _ = unit
fmap-ctx {Γ , A} (γ , a) p = fmap-ctx γ p , fmap-ty {A} a p
```