Categorial Grammar in Agda

Pepijn Kokke

March 26, 2015

Abstract

```
module main (Univ : Set ℓ) where

open import Relation.Binary.PropositionalEquality using (_≡_; refl)



infixr 20 _\_
infixl 20 _/_
infixl 25 _⊘_
infixr 25 _⦸_
infixr 30 _⊗_
infixr 30 _⊕_

data Type : Set ℓ where
  el  : Univ → Type
  _⊗_ : Type → Type → Type
  _\_ : Type → Type → Type
  _/_ : Type → Type → Type
  _⊕_ : Type → Type → Type
  _⊘_ : Type → Type → Type
  _⦸_ : Type → Type → Type

data Judgement : Set ℓ where
  _⊢_ : Type → Type → Judgement

infix 1 LG_

data LG_ : Judgement → Set ℓ where
  ax  : LG el A ⊢ el A
    -- rules for residuation and monotonicity for (/ , ⊗ , \)
  m⊗  : LG A ⊢ B      → LG C ⊢ D → LG A ⊗ C ⊢ B ⊗ D
```

```
  m\   : LG A ⊢ B       → LG C ⊢ D → LG B \ C ⊢ A \ D
  m/   : LG A ⊢ B       → LG C ⊢ D → LG A / D ⊢ B / C
  r\⊗  : LG B ⊢ A \ C  → LG A ⊗ B ⊢ C
  r⊗\  : LG A ⊗ B ⊢ C → LG B ⊢ A \ C
  r/⊗  : LG A ⊢ C / B  → LG A ⊗ B ⊢ C
  r⊗/  : LG A ⊗ B ⊢ C → LG A ⊢ C / B
    -- rules for residuation and monotonicity for (⊘ , ⊕ , ⦸)
  m⊕   : LG A ⊢ B       → LG C ⊢ D → LG A ⊕ C ⊢ B ⊕ D
  m⦸   : LG C ⊢ D       → LG A ⊢ B → LG D ⦸ A ⊢ C ⦸ B
  m⊘   : LG A ⊢ B       → LG C ⊢ D → LG A ⊘ D ⊢ B ⊘ C
  r⦸⊕  : LG B ⦸ C ⊢ A → LG C ⊢ B ⊕ A
  r⊕⦸  : LG C ⊢ B ⊕ A → LG B ⦸ C ⊢ A
  r⊕⊘  : LG C ⊢ B ⊕ A → LG C ⊘ A ⊢ B
  r⊘⊕  : LG C ⊘ A ⊢ B → LG C ⊢ B ⊕ A
    -- grishin distributives
  d⦸/  : LG A ⊗ B ⊢ C ⊕ D → LG C ⦸ A ⊢ D / B
  d⦸\  : LG A ⊗ B ⊢ C ⊕ D → LG C ⦸ B ⊢ A \ D
  d⊘\  : LG A ⊗ B ⊢ C ⊕ D → LG B ⊘ D ⊢ A \ C
  d⊘/  : LG A ⊗ B ⊢ C ⊕ D → LG A ⊘ D ⊢ C / B

ax′ : LG A ⊢ A
ax′ = ax
ax′ = m⊗ ax′ ax′
ax′ = m⊘ ax′ ax′
ax′ = m⦸ ax′ ax′
ax′ = m⊕ ax′ ax′
ax′ = m/ ax′ ax′
ax′ = m\ ax′ ax′

appl-\′  : LG A ⊗ (A \ B) ⊢ B
appl-\′  = r\⊗ (m\ ax′ ax′)
appl-/′  : LG (B / A) ⊗ A ⊢ B
appl-/′  = r/⊗ (m/ ax′ ax′)
appl-⦸′  : LG B ⊢ A ⊕ (A ⦸ B)
appl-⦸′  = r⦸⊕ (m⦸ ax′ ax′)
appl-⊘′  : LG B ⊢ (B ⊘ A) ⊕ A
appl-⊘′  = r⊘⊕ (m⊘ ax′ ax′)

infix 1 LG_⋯_

data LG_⋯_ : (I J : Judgement) → Set ℓ where
  []     : LG J ⋯ J
    -- rules for residuation and monotonicity for (/ , ⊗ , \)
  r\⊗   : LG J ⋯ B ⊢ A \ C  → LG J ⋯ A ⊗ B ⊢ C
  r⊗\   : LG J ⋯ A ⊗ B ⊢ C → LG J ⋯ B ⊢ A \ C
  r/⊗   : LG J ⋯ A ⊢ C / B  → LG J ⋯ A ⊗ B ⊢ C
```

```
r⊗/   : LG J ⋯ A ⊗ B ⊢ C → LG J ⋯ A ⊢ C / B
m⊗^L  : LG J ⋯ A ⊢ B → LG       C ⊢ D → LG J ⋯ A ⊗ C ⊢ B ⊗ D
m⊗^R  : LG       A ⊢ B → LG J ⋯ C ⊢ D → LG J ⋯ A ⊗ C ⊢ B ⊗ D
m\^L  : LG J ⋯ A ⊢ B → LG       C ⊢ D → LG J ⋯ B \ C ⊢ A \ D
m\^R  : LG       A ⊢ B → LG J ⋯ C ⊢ D → LG J ⋯ B \ C ⊢ A \ D
m/^L  : LG J ⋯ A ⊢ B → LG       C ⊢ D → LG J ⋯ A / D ⊢ B / C
m/^R  : LG       A ⊢ B → LG J ⋯ C ⊢ D → LG J ⋯ A / D ⊢ B / C
  -- rules for residuation and monotonicity for (⊘ , ⊕ , ⦸)
r⦸⊕   : LG J ⋯ B ⦸ C ⊢ A → LG J ⋯ C ⊢ B ⊕ A
r⊕⦸   : LG J ⋯ C ⊢ B ⊕ A → LG J ⋯ B ⦸ C ⊢ A
r⊕⊘   : LG J ⋯ C ⊢ B ⊕ A → LG J ⋯ C ⊘ A ⊢ B
r⊘⊕   : LG J ⋯ C ⊘ A ⊢ B → LG J ⋯ C ⊢ B ⊕ A
m⊕^L  : LG J ⋯ A ⊢ B → LG       C ⊢ D → LG J ⋯ A ⊕ C ⊢ B ⊕ D
m⊕^R  : LG       A ⊢ B → LG J ⋯ C ⊢ D → LG J ⋯ A ⊕ C ⊢ B ⊕ D
m⦸^L  : LG J ⋯ C ⊢ D → LG       A ⊢ B → LG J ⋯ D ⦸ A ⊢ C ⦸ B
m⦸^R  : LG       C ⊢ D → LG J ⋯ A ⊢ B → LG J ⋯ D ⦸ A ⊢ C ⦸ B
m⊘^L  : LG J ⋯ A ⊢ B → LG       C ⊢ D → LG J ⋯ A ⊘ D ⊢ B ⊘ C
m⊘^R  : LG       A ⊢ B → LG J ⋯ C ⊢ D → LG J ⋯ A ⊘ D ⊢ B ⊘ C
  -- grishin distributives
d⦸/   : LG J ⋯ A ⊗ B ⊢ C ⊕ D → LG J ⋯ C ⦸ A ⊢ D / B
d⦸\   : LG J ⋯ A ⊗ B ⊢ C ⊕ D → LG J ⋯ C ⦸ B ⊢ A \ D
d⊘\   : LG J ⋯ A ⊗ B ⊢ C ⊕ D → LG J ⋯ B ⊘ D ⊢ A \ C
d⊘/   : LG J ⋯ A ⊗ B ⊢ C ⊕ D → LG J ⋯ A ⊘ D ⊢ C / B


_$_ : LG I ⋯ J → LG I → LG J
[] $ x = x
r\⊗   f   $ x = r\⊗ (f $ x)
r⊗\   f   $ x = r⊗\ (f $ x)
r/⊗   f   $ x = r/⊗ (f $ x)
r⊗/   f   $ x = r⊗/ (f $ x)
m⊗^L  f g $ x = m⊗ (f $ x) g
m⊗^R  f g $ x = m⊗ f (g $ x)
m\^L  f g $ x = m\ (f $ x) g
m\^R  f g $ x = m\ f (g $ x)
m/^L  f g $ x = m/ (f $ x) g
m/^R  f g $ x = m/ f (g $ x)
r⦸⊕   f   $ x = r⦸⊕ (f $ x)
r⊕⦸   f   $ x = r⊕⦸ (f $ x)
r⊕⊘   f   $ x = r⊕⊘ (f $ x)
r⊘⊕   f   $ x = r⊘⊕ (f $ x)
m⊕^L  f g $ x = m⊕ (f $ x) g
m⊕^R  f g $ x = m⊕ f (g $ x)
m⦸^L  f g $ x = m⦸ (f $ x) g
m⦸^R  f g $ x = m⦸ f (g $ x)
m⊘^L  f g $ x = m⊘ (f $ x) g
m⊘^R  f g $ x = m⊘ f (g $ x)
```

```
d⦸/  f   $ x  =  d⦸/ (f $ x)
d⦸\  f   $ x  =  d⦸\ (f $ x)
d⊘\  f   $ x  =  d⊘\ (f $ x)
d⊘/  f   $ x  =  d⊘/ (f $ x)


_∘_ : LG J ⋯ K → LG I ⋯ J → LG I ⋯ K
[] ∘ h  =  h
r\⊗  f   ∘ h  =  r\⊗ (f ∘ h)
r⊗\  f   ∘ h  =  r⊗\ (f ∘ h)
r/⊗  f   ∘ h  =  r/⊗ (f ∘ h)
r⊗/  f   ∘ h  =  r⊗/ (f ∘ h)
m⊗ᴸ f g ∘ h  =  m⊗ᴸ (f ∘ h) g
m⊗ᴿ f g ∘ h  =  m⊗ᴿ f (g ∘ h)
m\ᴸ  f g ∘ h  =  m\ᴸ (f ∘ h) g
m\ᴿ  f g ∘ h  =  m\ᴿ f (g ∘ h)
m/ᴸ  f g ∘ h  =  m/ᴸ (f ∘ h) g
m/ᴿ  f g ∘ h  =  m/ᴿ f (g ∘ h)
r⦸⊕  f   ∘ h  =  r⦸⊕ (f ∘ h)
r⊕⦸  f   ∘ h  =  r⊕⦸ (f ∘ h)
r⊕⊘  f   ∘ h  =  r⊕⊘ (f ∘ h)
r⊘⊕  f   ∘ h  =  r⊘⊕ (f ∘ h)
m⊕ᴸ f g ∘ h  =  m⊕ᴸ (f ∘ h) g
m⊕ᴿ f g ∘ h  =  m⊕ᴿ f (g ∘ h)
m⦸ᴸ f g ∘ h  =  m⦸ᴸ (f ∘ h) g
m⦸ᴿ f g ∘ h  =  m⦸ᴿ f (g ∘ h)
m⊘ᴸ f g ∘ h  =  m⊘ᴸ (f ∘ h) g
m⊘ᴿ f g ∘ h  =  m⊘ᴿ f (g ∘ h)
d⦸/  f   ∘ h  =  d⦸/ (f ∘ h)
d⦸\  f   ∘ h  =  d⦸\ (f ∘ h)
d⊘\  f   ∘ h  =  d⊘\ (f ∘ h)
d⊘/  f   ∘ h  =  d⊘/ (f ∘ h)


∘-def : (f : LG I ⋯ J) (g : LG J ⋯ K) (x : LG I) → g $ (f $ x) ≡ (g ∘ f) $ x
∘-def f []            x  =  refl
∘-def f (r\⊗ g)       x rewrite ∘-def f g x  =  refl
∘-def f (r⊗\ g)       x rewrite ∘-def f g x  =  refl
∘-def f (r/⊗ g)       x rewrite ∘-def f g x  =  refl
∘-def f (r⊗/ g)       x rewrite ∘-def f g x  =  refl
∘-def f (m⊗ᴸ g h)    x rewrite ∘-def f g x  =  refl
∘-def f (m⊗ᴿ g h)    x rewrite ∘-def f h x  =  refl
∘-def f (m\ᴸ g h)     x rewrite ∘-def f g x  =  refl
∘-def f (m\ᴿ g h)     x rewrite ∘-def f h x  =  refl
∘-def f (m/ᴸ g h)     x rewrite ∘-def f g x  =  refl
∘-def f (m/ᴿ g h)     x rewrite ∘-def f h x  =  refl
∘-def f (r⦸⊕ g)       x rewrite ∘-def f g x  =  refl
∘-def f (r⊕⦸ g)       x rewrite ∘-def f g x  =  refl
∘-def f (r⊕⊘ g)       x rewrite ∘-def f g x  =  refl
```

```
∘-def f (r⊘⊕ g)     x rewrite ∘-def f g x  =  refl
∘-def f (m⊕^L g h) x rewrite ∘-def f g x  =  refl
∘-def f (m⊕^R g h) x rewrite ∘-def f h x  =  refl
∘-def f (m⦸^L g h) x rewrite ∘-def f g x  =  refl
∘-def f (m⦸^R g h) x rewrite ∘-def f h x  =  refl
∘-def f (m⊘^L g h) x rewrite ∘-def f g x  =  refl
∘-def f (m⊘^R g h) x rewrite ∘-def f h x  =  refl
∘-def f (d⦸/ g)     x rewrite ∘-def f g x  =  refl
∘-def f (d⦸\ g)     x rewrite ∘-def f g x  =  refl
∘-def f (d⊘\ g)     x rewrite ∘-def f g x  =  refl
∘-def f (d⊘/ g)     x rewrite ∘-def f g x  =  refl
```

```
data Context : Set ℓ where
  []     : Context
  _⊗>_   : Type    → Context → Context
  _\>_   : Type    → Context → Context
  _/>_   : Type    → Context → Context
  _<⊗_   : Context → Type    → Context
  _<\_   : Context → Type    → Context
  _</_   : Context → Type    → Context
  _⊕>_   : Type    → Context → Context
  _⊘>_   : Type    → Context → Context
  _⦸>_   : Type    → Context → Context
  _<⊕_   : Context → Type    → Context
  _<⊘_   : Context → Type    → Context
  _<⦸_   : Context → Type    → Context
```

```
_[_] : Context → Type → Type
[]        [A]  =  A
(B ⊗> C) [A]  =  B ⊗ (C [A])
(B \> C) [A]  =  B \ (C [A])
(B /> C) [A]  =  B / (C [A])
(B ⊕> C) [A]  =  B ⊕ (C [A])
(B ⦸> C) [A]  =  B ⦸ (C [A])
(B ⊘> C) [A]  =  B ⊘ (C [A])
(C <⊗ B) [A]  =  (C [A]) ⊗ B
(C <\ B) [A]  =  (C [A]) \ B
(C </ B) [A]  =  (C [A]) / B
(C <⊕ B) [A]  =  (C [A]) ⊕ B
(C <⦸ B) [A]  =  (C [A]) ⦸ B
(C <⊘ B) [A]  =  (C [A]) ⊘ B
```

```
_⟨_⟩ : Context → Context → Context
[]        ⟨ A ⟩  =  A
(B ⊗> C) ⟨ A ⟩  =  B ⊗> (C ⟨ A ⟩)
(B \> C) ⟨ A ⟩  =  B \> (C ⟨ A ⟩)
(B /> C) ⟨ A ⟩  =  B /> (C ⟨ A ⟩)
(B ⊕> C) ⟨ A ⟩  =  B ⊕> (C ⟨ A ⟩)
```

```
(B ⦸> C) ⟨ A ⟩ = B ⦸> (C ⟨ A ⟩)
(B ⊘> C) ⟨ A ⟩ = B ⊘> (C ⟨ A ⟩)
(C <⊗ B) ⟨ A ⟩ = (C ⟨ A ⟩) <⊗ B
(C <\ B) ⟨ A ⟩ = (C ⟨ A ⟩) <\ B
(C </ B) ⟨ A ⟩ = (C ⟨ A ⟩) </ B
(C <⊕ B) ⟨ A ⟩ = (C ⟨ A ⟩) <⊕ B
(C <⦸ B) ⟨ A ⟩ = (C ⟨ A ⟩) <⦸ B
(C <⊘ B) ⟨ A ⟩ = (C ⟨ A ⟩) <⊘ B


data Polarity : Set where
  + : Polarity
  $-$ : Polarity


data Polarised (p : Polarity) : Polarity → Context → Set ℓ where
  [] : Polarised p p []
  _⊗>_ : (A : Type) (B⁺ : Polarised p + B) → Polarised p + (A ⊗> B)
  _⦸>_ : (A : Type) (B⁺ : Polarised p + B) → Polarised p + (A ⦸> B)
  _⊘>_ : (A : Type) (B⁻ : Polarised p $-$ B) → Polarised p + (A ⊘> B)
  _⊕>_ : (A : Type) (B⁻ : Polarised p $-$ B) → Polarised p $-$ (A ⊕> B)
  _\>_ : (A : Type) (B⁻ : Polarised p $-$ B) → Polarised p $-$ (A \> B)
  _/>_ : (A : Type) (B⁺ : Polarised p + B) → Polarised p $-$ (A /> B)
  _<⊗_ : (A⁺ : Polarised p + A) (B : Type) → Polarised p + (A <⊗ B)
  _<⦸_ : (A⁻ : Polarised p $-$ A) (B : Type) → Polarised p + (A <⦸ B)
  _<⊘_ : (A⁺ : Polarised p + A) (B : Type) → Polarised p + (A <⊘ B)
  _<⊕_ : (A⁻ : Polarised p $-$ A) (B : Type) → Polarised p $-$ (A <⊕ B)
  _<\_ : (A⁺ : Polarised p + A) (B : Type) → Polarised p $-$ (A <\ B)
  _</_ : (A⁻ : Polarised p $-$ A) (B : Type) → Polarised p $-$ (A </ B)


_ [_] ᴾ : Polarised p₁ p₂ A → Type → Type
[] [A] ᴾ = A
(B ⊗> C) [A] ᴾ = B ⊗ (C [A] ᴾ)
(B \> C) [A] ᴾ = B \ (C [A] ᴾ)
(B /> C) [A] ᴾ = B / (C [A] ᴾ)
(B ⊕> C) [A] ᴾ = B ⊕ (C [A] ᴾ)
(B ⦸> C) [A] ᴾ = B ⦸ (C [A] ᴾ)
(B ⊘> C) [A] ᴾ = B ⊘ (C [A] ᴾ)
(C <⊗ B) [A] ᴾ = (C [A] ᴾ) ⊗ B
(C <\ B) [A] ᴾ = (C [A] ᴾ) \ B
(C </ B) [A] ᴾ = (C [A] ᴾ) / B
(C <⊕ B) [A] ᴾ = (C [A] ᴾ) ⊕ B
(C <⦸ B) [A] ᴾ = (C [A] ᴾ) ⦸ B
(C <⊘ B) [A] ᴾ = (C [A] ᴾ) ⊘ B


_⟨_⟩ᴾ : Polarised p₂ p₃ A → Polarised p₁ p₂ B → Polarised p₁ p₃ (A ⟨ B ⟩)
[] ⟨ A ⟩ᴾ = A
```

```
(B ⊗> C) ⟨ A ⟩^P  =  B ⊗> (C ⟨ A ⟩^P)
(B \> C) ⟨ A ⟩^P  =  B \> (C ⟨ A ⟩^P)
(B /> C) ⟨ A ⟩^P  =  B /> (C ⟨ A ⟩^P)
(B ⊕> C) ⟨ A ⟩^P  =  B ⊕> (C ⟨ A ⟩^P)
(B ⦸> C) ⟨ A ⟩^P  =  B ⦸> (C ⟨ A ⟩^P)
(B ⊘> C) ⟨ A ⟩^P  =  B ⊘> (C ⟨ A ⟩^P)
(C <⊗ B) ⟨ A ⟩^P  =  (C ⟨ A ⟩^P) <⊗ B
(C <\ B) ⟨ A ⟩^P  =  (C ⟨ A ⟩^P) <\ B
(C </ B) ⟨ A ⟩^P  =  (C ⟨ A ⟩^P) </ B
(C <⊕ B) ⟨ A ⟩^P  =  (C ⟨ A ⟩^P) <⊕ B
(C <⦸ B) ⟨ A ⟩^P  =  (C ⟨ A ⟩^P) <⦸ B
(C <⊘ B) ⟨ A ⟩^P  =  (C ⟨ A ⟩^P) <⊘ B

data Context^J : Set ℓ where
  _<⊢_ : Context → Type → Context^J
  _⊢>_ : Type → Context → Context^J

_[_]^J : Context^J → Type → Judgement
(A <⊢ B) [C]^J = A [C] ⊢ B
(A ⊢> B) [C]^J = A ⊢ B [C]

_⟨_⟩^J : Context^J → Context → Context^J
_⟨_⟩^J (A <⊢ B) C = A ⟨ C ⟩ <⊢ B
_⟨_⟩^J (A ⊢> B) C = A ⊢> B ⟨ C ⟩

data Polarised^J (p : Polarity) : Context^J → Set ℓ where
  _<⊢_ : (A⁺ : Polarised p + A) (B : Type) → Polarised^J p (A <⊢ B)
  _⊢>_ : ∀ (A : Type) (B⁻ : Polarised p $-$ B) → Polarised^J p (A ⊢> B)

module el where

data Origin (J⁺ : Polarised^J + J) (f : LG J [el B]^J) : Set ℓ where
  origin : (f′ : LG G ⊢ el B ⋯ J [G]^J)
    → (pr : f ≡ f′ $ ax)
    → Origin J⁺ f

mutual
  viewOrigin : (J⁺ : Polarised^J + J) (f : LG J [el B]^J) → Origin J⁺ f
  viewOrigin ([] <⊢ ._) ax          = origin [] refl
  viewOrigin ([] <⊢ ._) (r⊗\ f)     = go ((_ ⊗> []) <⊢ _) f     (r⊗\ [])
  viewOrigin ([] <⊢ ._) (r⊗/ f)     = go (([] <⊗ _) <⊢ _) f     (r⊗/ [])
  viewOrigin ([] <⊢ ._) (r⦸⊕ f)     = go ((_ ⦸> []) <⊢ _) f     (r⦸⊕ [])
  viewOrigin ([] <⊢ ._) (r⊘⊕ f)     = go (([] <⊘ _) <⊢ _) f     (r⊘⊕ [])
```

```
viewOrigin ((A ⊗> B) <⊢ ._) (m⊗ f₁ f₂)  =  go (B <⊢ _)          f₂ (m⊗ᴿ f₁ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r\⊗ f)   =  go (B <⊢ _)           f          (r\⊗ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A ⊗> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r/⊗ f)   =  go (_ ⊢> (_ /> B))  f          (r/⊗ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⊗/ f)   =  go (((A ⊗> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⦸⊕ f)   =  go ((_ ⦸> (A ⊗> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⊘⊕ f)   =  go (((A ⊗> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A ⦸> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊗/ f)   =  go (((A ⦸> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⦸> B) <⊢ ._) (m⦸ f₁ f₂)  =  go (B <⊢ _)          f₂ (m⦸ᴿ f₁ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⦸⊕ f)   =  go ((_ ⦸> (A ⦸> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊕⦸ f)   =  go (B <⊢ _)           f          (r⊕⦸ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊘⊕ f)   =  go (((A ⦸> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (d⦸/ f)   =  go ((B <⊗ _) <⊢ _) f          (d⦸/ [])
viewOrigin ((A ⦸> B) <⊢ ._) (d⦸\ f)   =  go ((_ ⊗> B) <⊢ _) f          (d⦸\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A ⊘> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗/ f)   =  go (((A ⊘> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⊘> B) <⊢ ._) (m⊘ f₁ f₂)  =  go (_ ⊢> B)          f₂ (m⊘ᴿ f₁ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⦸⊕ f)   =  go ((_ ⦸> (A ⊘> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊕⊘ f)   =  go (_ ⊢> (_ ⊕> B)) f          (r⊕⊘ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊘⊕ f)   =  go (((A ⊘> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘\ f)   =  go (_ ⊢> (_ ⊕> B)) f          (d⊘\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘/ f)   =  go (_ ⊢> (_ ⊕> B)) f          (d⊘/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (m⊗ f₁ f₂)  =  go (A <⊢ _)          f₁ (m⊗ᴸ [] f₂)
viewOrigin ((A <⊗ B) <⊢ ._) (r\⊗ f)   =  go (_ ⊢> (A <\ _))  f          (r\⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A <⊗ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r/⊗ f)   =  go (A <⊢ _)           f          (r/⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗/ f)   =  go (((A <⊗ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⦸⊕ f)   =  go ((_ ⦸> (A <⊗ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊘⊕ f)   =  go (((A <⊗ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A <⦸ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗/ f)   =  go (((A <⦸ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (m⦸ f₁ f₂)  =  go (_ ⊢> A)          f₁ (m⦸ᴸ [] f₂)
viewOrigin ((A <⦸ B) <⊢ ._) (r⦸⊕ f)   =  go ((_ ⦸> (A <⦸ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊕⦸ f)   =  go (_ ⊢> (A <⊕ _)) f          (r⊕⦸ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊘⊕ f)   =  go (((A <⦸ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸/ f)   =  go (_ ⊢> (A <⊕ _)) f          (d⦸/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸\ f)   =  go (_ ⊢> (A <⊕ _)) f          (d⦸\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A <⊘ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗/ f)   =  go (((A <⊘ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊘ B) <⊢ ._) (m⊘ f₁ f₂)  =  go (A <⊢ _)          f₁ (m⊘ᴸ [] f₂)
viewOrigin ((A <⊘ B) <⊢ ._) (r⦸⊕ f)   =  go ((_ ⦸> (A <⊘ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊕⊘ f)   =  go (A <⊢ _)           f          (r⊕⊘ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊘⊕ f)   =  go (((A <⊘ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘\ f)   =  go ((_ ⊗> A) <⊢ _) f          (d⊘\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘/ f)   =  go ((A <⊗ _) <⊢ _) f          (d⊘/ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r\⊗ f)   =  go (_ ⊢> (_ \> (A ⊕> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r/⊗ f)   =  go (_ ⊢> ((A ⊕> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (m⊕ f₁ f₂)  =  go (_ ⊢> B)          f₂ (m⊕ᴿ f₁ [])
```

```
viewOrigin (._ ⊢> (A ⊕> B)) (r⦸⊕ f)  = go (_ ⊢> B)              f (r⦸⊕ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⦸ f)  = go (_ ⊢> (_ ⊕> (A ⊕> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⊘ f)  = go (_ ⊢> ((A ⊕> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊘⊕ f)  = go ((_ ⊘> B) <⊢ _) f       (r⊘⊕ [])
viewOrigin (._ ⊢> (A \> B)) (m\  f1 f2) = go (_ ⊢> B)           f2 (m\^R f1 [])
viewOrigin (._ ⊢> (A \> B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A \> B))) f (r\⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊗\ f)   = go (_ ⊢> B)             f  (r⊗\ [])
viewOrigin (._ ⊢> (A \> B)) (r/⊗ f)   = go (_ ⊢> ((A \> B) </ _)) f (r/⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A \> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⊘ f)   = go (_ ⊢> ((A \> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A \> B)) (d⦸\ f)   = go (_ ⊢> (_ ⊕> B)) f       (d⦸\ [])
viewOrigin (._ ⊢> (A \> B)) (d⊘\ f)   = go (_ ⊢> (B <⊕ _)) f       (d⊘\ [])
viewOrigin (._ ⊢> (A /> B)) (m/  f1 f2) = go (B <⊢ _)           f2 (m/^R f1 [])
viewOrigin (._ ⊢> (A /> B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A /> B))) f (r\⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r/⊗ f)   = go (_ ⊢> ((A /> B) </ _)) f (r/⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r⊗/ f)   = go ((_ ⊗> B) <⊢ _) f       (r⊗/ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A /> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⊘ f)   = go (_ ⊢> ((A /> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A /> B)) (d⦸/ f)   = go ((_ ⊗> B) <⊢ _) f       (d⦸/ [])
viewOrigin (._ ⊢> (A /> B)) (d⊘/ f)   = go ((_ ⊗> B) <⊢ _) f       (d⊘/ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A <⊕ B))) f (r\⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r/⊗ f)   = go (_ ⊢> ((A <⊕ B) </ _)) f (r/⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (m⊕ f1 f2) = go (_ ⊢> A)            f1 (m⊕^L [] f2)
viewOrigin (._ ⊢> (A <⊕ B)) (r⦸⊕ f)  = go ((A <⦸ _) <⊢ _) f       (r⦸⊕ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⦸ f)  = go (_ ⊢> (_ ⊕> (A <⊕ B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⊘ f)  = go (_ ⊢> ((A <⊕ B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊘⊕ f)  = go (_ ⊢> A)              f (r⊘⊕ [])
viewOrigin (._ ⊢> (A <\ B)) (m\  f1 f2) = go (A <⊢ _)           f1 (m\^L [] f2)
viewOrigin (._ ⊢> (A <\ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A <\ B))) f (r\⊗ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊗\ f)   = go ((A <⊗ _) <⊢ _) f       (r⊗\ [])
viewOrigin (._ ⊢> (A <\ B)) (r/⊗ f)   = go (_ ⊢> ((A <\ B) </ _)) f (r/⊗ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A <\ B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊕⊘ f)   = go (_ ⊢> ((A <\ B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A <\ B)) (d⦸\ f)   = go ((A <⊗ _) <⊢ _) f       (d⦸\ [])
viewOrigin (._ ⊢> (A <\ B)) (d⊘\ f)   = go ((A <⊗ _) <⊢ _) f       (d⊘\ [])
viewOrigin (._ ⊢> (A </ B)) (m/  f1 f2) = go (_ ⊢> A)           f1 (m/^L [] f2)
viewOrigin (._ ⊢> (A </ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A </ B))) f (r\⊗ [])
viewOrigin (._ ⊢> (A </ B)) (r/⊗ f)   = go (_ ⊢> ((A </ B) </ _)) f (r/⊗ [])
viewOrigin (._ ⊢> (A </ B)) (r⊗/ f)   = go (_ ⊢> A)             f  (r⊗/ [])
viewOrigin (._ ⊢> (A </ B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A </ B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A </ B)) (r⊕⊘ f)   = go (_ ⊢> ((A </ B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A </ B)) (d⦸/ f)   = go (_ ⊢> (_ ⊕> A)) f       (d⦸/ [])
viewOrigin (._ ⊢> (A </ B)) (d⊘/ f)   = go (_ ⊢> (A <⊕ _)) f       (d⊘/ [])
private
  go : (I+ : Polarised^J + I) (f : LG I [el B] ^J)
     → (g : LG I [G] ^J ⋯ J [G] ^J)
     → Origin J+ (g $ f)
  go I+ f g with viewOrigin I+ f
  ... | origin f′ pr rewrite pr | ∘-def f′ g ax = origin (g ∘ f′) refl
```

module ⊗ **where**

```
data Origin (J⁻ : Polarised^J $-$ J) (f : LG J [B ⊗ C] ^J) : Set ℓ where
  origin : (h₁ : LG E ⊢ B) (h₂ : LG F ⊢ C)
    → (f′ : LG E ⊗ F ⊢ G ⋯ J [G] ^J)
    → (pr : f ≡ f′ $ m⊗ h₁ h₂)
    → Origin J⁻ f
```

```
mutual
  viewOrigin : (J⁻ : Polarised^J $-$ J) (f : LG J [B ⊗ C] ^J) → Origin J⁻ f
  viewOrigin (._ ⊢> []) (m⊗        f₁ f₂) = origin f₁ f₂ [] refl
  viewOrigin (._ ⊢> []) (r\⊗ f)          = go (_ ⊢> (_ \> [])) f         (r\⊗ [])
  viewOrigin (._ ⊢> []) (r/⊗ f)          = go (_ ⊢> ([] </ _)) f         (r/⊗ [])
  viewOrigin (._ ⊢> []) (r⊕⦸ f)          = go (_ ⊢> (_ ⊕> [])) f         (r⊕⦸ [])
  viewOrigin (._ ⊢> []) (r⊕⊘ f)          = go (_ ⊢> ([] <⊕ _)) f         (r⊕⊘ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (m⊗ f₁ f₂) = go (B <⊢ _)         f₂ (m⊗^R f₁ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r\⊗ f)  = go (B <⊢ _)          f        (r\⊗ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A ⊗> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r/⊗ f)  = go (_ ⊢> (_ /> B))  f        (r/⊗ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊗/ f)  = go (((A ⊗> B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⦸⊕ f)  = go ((_ ⦸> (A ⊗> B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊘⊕ f)  = go (((A ⊗> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A ⦸> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊗/ f)  = go (((A ⦸> B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (m⦸ f₁ f₂) = go (B <⊢ _)         f₂ (m⦸^R f₁ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⦸⊕ f)  = go ((_ ⦸> (A ⦸> B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊕⦸ f)  = go (B <⊢ _)          f        (r⊕⦸ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊘⊕ f)  = go (((A ⦸> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (d⦸/ f)  = go ((B <⊗ _) <⊢ _) f         (d⦸/ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (d⦸\ f)  = go ((_ ⊗> B) <⊢ _) f         (d⦸\ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A ⊘> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⊗/ f)  = go (((A ⊘> B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (m⊘ f₁ f₂) = go (_ ⊢> B)         f₂ (m⊘^R f₁ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⦸⊕ f)  = go ((_ ⦸> (A ⊘> B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⊕⊘ f)  = go (_ ⊢> (_ ⊕> B)) f         (r⊕⊘ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⊘⊕ f)  = go (((A ⊘> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (d⊘\ f)  = go (_ ⊢> (_ ⊕> B)) f         (d⊘\ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (d⊘/ f)  = go (_ ⊢> (_ ⊕> B)) f         (d⊘/ [])
  viewOrigin ((A <⊗ B) <⊢ ._) (m⊗ f₁ f₂) = go (A <⊢ _)         f₁ (m⊗^L [] f₂)
  viewOrigin ((A <⊗ B) <⊢ ._) (r\⊗ f)  = go (_ ⊢> (A <\ _))  f        (r\⊗ [])
  viewOrigin ((A <⊗ B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A <⊗ B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A <⊗ B) <⊢ ._) (r/⊗ f)  = go (A <⊢ _)          f        (r/⊗ [])
  viewOrigin ((A <⊗ B) <⊢ ._) (r⊗/ f)  = go (((A <⊗ B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A <⊗ B) <⊢ ._) (r⦸⊕ f)  = go ((_ ⦸> (A <⊗ B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A <⊗ B) <⊢ ._) (r⊘⊕ f)  = go (((A <⊗ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A <⦸ B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A <⦸ B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A <⦸ B) <⊢ ._) (r⊗/ f)  = go (((A <⦸ B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A <⦸ B) <⊢ ._) (m⦸ f₁ f₂) = go (_ ⊢> A)         f₁ (m⦸^L [] f₂)
```

```
viewOrigin ((A <⦸ B) <⊢ ._) (r⦸⊕ f)  =  go ((_ ⦸> (A <⦸ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊕⦸ f)  =  go (_ ⊢> (A <⊕ _)) f         (r⊕⦸ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊘⊕ f)  =  go (((A <⦸ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸/ f)  =  go (_ ⊢> (A <⊕ _)) f         (d⦸/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸\ f)  =  go (_ ⊢> (A <⊕ _)) f         (d⦸\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗\ f)  =  go ((_ ⊗> (A <⊘ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗/ f)  =  go (((A <⊘ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊘ B) <⊢ ._) (m⊘ f1 f2)  =  go (A <⊢ _)      f1 (m⊘^L [] f2)
viewOrigin ((A <⊘ B) <⊢ ._) (r⦸⊕ f)  =  go ((_ ⦸> (A <⊘ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊕⊘ f)  =  go (A <⊢ _)           f      (r⊕⊘ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊘⊕ f)  =  go (((A <⊘ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘\ f)  =  go ((_ ⊗> A) <⊢ _) f         (d⊘\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘/ f)  =  go ((A <⊗ _) <⊢ _) f         (d⊘/ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r\⊗ f)  =  go (_ ⊢> (_ \> (A ⊕> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r/⊗ f)  =  go (_ ⊢> ((A ⊕> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (m⊕ f1 f2)  =  go (_ ⊢> B)      f2 (m⊕^R f1 [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⦸⊕ f)  =  go (_ ⊢> B)           f      (r⦸⊕ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⦸ f)  =  go (_ ⊢> (_ ⊕> (A ⊕> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⊘ f)  =  go (_ ⊢> ((A ⊕> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊘⊕ f)  =  go ((_ ⊘> B) <⊢ _) f         (r⊘⊕ [])
viewOrigin (._ ⊢> (A \> B)) (m\  f1 f2)  =  go (_ ⊢> B)      f2 (m\^R f1 [])
viewOrigin (._ ⊢> (A \> B)) (r\⊗ f)   =  go (_ ⊢> (_ \> (A \> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊗\ f)   =  go (_ ⊢> B)           f      (r⊗\ [])
viewOrigin (._ ⊢> (A \> B)) (r/⊗ f)   =  go (_ ⊢> ((A \> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⦸ f)   =  go (_ ⊢> (_ ⊕> (A \> B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⊘ f)   =  go (_ ⊢> ((A \> B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A \> B)) (d⦸\ f)   =  go (_ ⊢> (_ ⊕> B)) f         (d⦸\ [])
viewOrigin (._ ⊢> (A \> B)) (d⊘\ f)   =  go (_ ⊢> (B <⊕ _)) f         (d⊘\ [])
viewOrigin (._ ⊢> (A /> B)) (m/  f1 f2)  =  go (B <⊢ _)      f2 (m/^R f1 [])
viewOrigin (._ ⊢> (A /> B)) (r\⊗ f)   =  go (_ ⊢> (_ \> (A /> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r/⊗ f)   =  go (_ ⊢> ((A /> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r⊗/ f)   =  go ((_ ⊗> B) <⊢ _) f         (r⊗/ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⦸ f)   =  go (_ ⊢> (_ ⊕> (A /> B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⊘ f)   =  go (_ ⊢> ((A /> B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A /> B)) (d⦸/ f)   =  go ((_ ⊗> B) <⊢ _) f         (d⦸/ [])
viewOrigin (._ ⊢> (A /> B)) (d⊘/ f)   =  go ((_ ⊗> B) <⊢ _) f         (d⊘/ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r\⊗ f)   =  go (_ ⊢> (_ \> (A <⊕ B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r/⊗ f)   =  go (_ ⊢> ((A <⊕ B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (m⊕ f1 f2)  =  go (_ ⊢> A)      f1 (m⊕^L [] f2)
viewOrigin (._ ⊢> (A <⊕ B)) (r⦸⊕ f)  =  go ((A <⦸ _) <⊢ _) f         (r⦸⊕ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⦸ f)  =  go (_ ⊢> (_ ⊕> (A <⊕ B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⊘ f)  =  go (_ ⊢> ((A <⊕ B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊘⊕ f)  =  go (_ ⊢> A)           f      (r⊘⊕ [])
viewOrigin (._ ⊢> (A <\ B)) (m\  f1 f2)  =  go (A <⊢ _)      f1 (m\^L [] f2)
viewOrigin (._ ⊢> (A <\ B)) (r\⊗ f)   =  go (_ ⊢> (_ \> (A <\ B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊗\ f)   =  go ((A <⊗ _) <⊢ _) f         (r⊗\ [])
viewOrigin (._ ⊢> (A <\ B)) (r/⊗ f)   =  go (_ ⊢> ((A <\ B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊕⦸ f)   =  go (_ ⊢> (_ ⊕> (A <\ B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊕⊘ f)   =  go (_ ⊢> ((A <\ B) <⊕ _)) f  (r⊕⊘ [])
```

```
  viewOrigin (._ ⊢> (A <\ B)) (d⦸\ f)   = go ((A <⊗ _) <⊢ _) f           (d⦸\ [])
  viewOrigin (._ ⊢> (A <\ B)) (d⊘\ f)   = go ((A <⊗ _) <⊢ _) f           (d⊘\ [])
  viewOrigin (._ ⊢> (A </ B)) (m/  f₁ f₂) = go (_ ⊢> A)        f₁ (m/ᴸ [] f₂)
  viewOrigin (._ ⊢> (A </ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A </ B))) f  (r\⊗ [])
  viewOrigin (._ ⊢> (A </ B)) (r/⊗ f)   = go (_ ⊢> ((A </ B) </ _)) f  (r/⊗ [])
  viewOrigin (._ ⊢> (A </ B)) (r⊗/ f)   = go (_ ⊢> A)           f      (r⊗/ [])
  viewOrigin (._ ⊢> (A </ B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A </ B))) f  (r⊕⦸ [])
  viewOrigin (._ ⊢> (A </ B)) (r⊕⊘ f)   = go (_ ⊢> ((A </ B) <⊕ _)) f  (r⊕⊘ [])
  viewOrigin (._ ⊢> (A </ B)) (d⦸/ f)   = go (_ ⊢> (_ ⊕> A)) f         (d⦸/ [])
  viewOrigin (._ ⊢> (A </ B)) (d⊘/ f)   = go (_ ⊢> (A <⊕ _)) f         (d⊘/ [])
  private
    go : (I⁻ : Polarisedᴶ $-$ I) (f : LG I [B ⊗ C] ᴶ)
       → (g : LG I [G] ᴶ ⋯ J [G] ᴶ)
       → Origin J⁻ (g $ f)
    go I⁻ f g with viewOrigin I⁻ f
    ... | origin h₁ h₂ f′ pr rewrite pr | ∘-def f′ g (m⊗ h₁ h₂) = origin h₁ h₂ (g ∘ f′) refl


module \ where


data Origin (J⁺ : Polarisedᴶ + J) (f : LG J [B \ C] ᴶ) : Set ℓ where
  origin : (h₁ : LG E ⊢ B) (h₂ : LG C ⊢ F)
    → (f′ : LG G ⊢ E \ F ⋯ J [G] ᴶ)
    → (pr : f ≡ f′ $ m\ h₁ h₂)
    → Origin J⁺ f


mutual
  viewOrigin : (J⁺ : Polarisedᴶ + J) (f : LG J [B \ C] ᴶ) → Origin J⁺ f
  viewOrigin ([] <⊢ ._) (m\          f₁ f₂) = origin f₁ f₂ [] refl
  viewOrigin ([] <⊢ ._) (r⊗\ f)             = go ((_ ⊗> []) <⊢ _) f       (r⊗\ [])
  viewOrigin ([] <⊢ ._) (r⊗/ f)             = go (([] <⊗ _) <⊢ _) f       (r⊗/ [])
  viewOrigin ([] <⊢ ._) (r⦸⊕ f)             = go ((_ ⦸> []) <⊢ _) f       (r⦸⊕ [])
  viewOrigin ([] <⊢ ._) (r⊘⊕ f)             = go (([] <⊘ _) <⊢ _) f       (r⊘⊕ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (m⊗ f₁ f₂) = go (B <⊢ _)          f₂ (m⊗ᴿ f₁ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r\⊗ f)  = go (B <⊢ _)           f      (r\⊗ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A ⊗> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r/⊗ f)  = go (_ ⊢> (_ /> B))  f      (r/⊗ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊗/ f)  = go (((A ⊗> B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⦸⊕ f)  = go ((_ ⦸> (A ⊗> B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊘⊕ f)  = go (((A ⊗> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A ⦸> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊗/ f)  = go (((A ⦸> B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (m⦸ f₁ f₂) = go (B <⊢ _)          f₂ (m⦸ᴿ f₁ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⦸⊕ f)  = go ((_ ⦸> (A ⦸> B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊕⦸ f)  = go (B <⊢ _)           f      (r⊕⦸ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊘⊕ f)  = go (((A ⦸> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (d⦸/ f)  = go ((B <⊗ _) <⊢ _) f       (d⦸/ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (d⦸\ f)  = go ((_ ⊗> B) <⊢ _) f       (d⦸\ [])
```

```
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A ⊘> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗/ f)   = go (((A ⊘> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⊘> B) <⊢ ._) (m⊘ f_1 f_2) = go (_ ⊢> B)        f_2 (m⊘^R f_1 [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A ⊘> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊕⊘ f)   = go (_ ⊢> (_ ⊕> B)) f          (r⊕⊘ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊘⊕ f)   = go (((A ⊘> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘\ f)   = go (_ ⊢> (_ ⊕> B)) f          (d⊘\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘/ f)   = go (_ ⊢> (_ ⊕> B)) f          (d⊘/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (m⊗ f_1 f_2) = go (A <⊢ _)        f_1 (m⊗^L [] f_2)
viewOrigin ((A <⊗ B) <⊢ ._) (r\⊗ f)   = go (_ ⊢> (A <\ _))  f          (r\⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A <⊗ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r/⊗ f)   = go (A <⊢ _)         f          (r/⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗/ f)   = go (((A <⊗ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A <⊗ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊘⊕ f)   = go (((A <⊗ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A <⦸ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗/ f)   = go (((A <⦸ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (m⦸ f_1 f_2) = go (_ ⊢> A)        f_1 (m⦸^L [] f_2)
viewOrigin ((A <⦸ B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A <⦸ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊕⦸ f)   = go (_ ⊢> (A <⊕ _)) f          (r⊕⦸ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊘⊕ f)   = go (((A <⦸ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸/ f)   = go (_ ⊢> (A <⊕ _)) f          (d⦸/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸\ f)   = go (_ ⊢> (A <⊕ _)) f          (d⦸\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A <⊘ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗/ f)   = go (((A <⊘ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊘ B) <⊢ ._) (m⊘ f_1 f_2) = go (A <⊢ _)        f_1 (m⊘^L [] f_2)
viewOrigin ((A <⊘ B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A <⊘ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊕⊘ f)   = go (A <⊢ _)         f          (r⊕⊘ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊘⊕ f)   = go (((A <⊘ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘\ f)   = go ((_ ⊗> A) <⊢ _) f          (d⊘\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘/ f)   = go ((A <⊗ _) <⊢ _) f          (d⊘/ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A ⊕> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r/⊗ f)   = go (_ ⊢> ((A ⊕> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (m⊕ f_1 f_2) = go (_ ⊢> B)        f_2 (m⊕^R f_1 [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⦸⊕ f)   = go (_ ⊢> B)         f          (r⦸⊕ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A ⊕> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⊘ f)   = go (_ ⊢> ((A ⊕> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊘⊕ f)   = go ((_ ⊘> B) <⊢ _) f          (r⊘⊕ [])
viewOrigin (._ ⊢> (A \> B)) (m\  f_1 f_2) = go (_ ⊢> B)       f_2 (m\^R f_1 [])
viewOrigin (._ ⊢> (A \> B)) (r\⊗ f)    = go (_ ⊢> (_ \> (A \> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊗\ f)    = go (_ ⊢> B)         f          (r⊗\ [])
viewOrigin (._ ⊢> (A \> B)) (r/⊗ f)    = go (_ ⊢> ((A \> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⦸ f)    = go (_ ⊢> (_ ⊕> (A \> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⊘ f)    = go (_ ⊢> ((A \> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A \> B)) (d⦸\ f)    = go (_ ⊢> (_ ⊕> B)) f          (d⦸\ [])
viewOrigin (._ ⊢> (A \> B)) (d⊘\ f)    = go (_ ⊢> (B <⊕ _)) f          (d⊘\ [])
viewOrigin (._ ⊢> (A /> B)) (m/  f_1 f_2) = go (B <⊢ _)       f_2 (m/^R f_1 [])
viewOrigin (._ ⊢> (A /> B)) (r\⊗ f)    = go (_ ⊢> (_ \> (A /> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r/⊗ f)    = go (_ ⊢> ((A /> B) </ _)) f  (r/⊗ [])
```

```
    viewOrigin (._ ⊢> (A /> B)) (r⊗/ f)    = go ((_ ⊗> B) <⊢ _) f            (r⊗/ [])
    viewOrigin (._ ⊢> (A /> B)) (r⊕⦸ f)    = go (_ ⊢> (_ ⊕> (A /> B))) f  (r⊕⦸ [])
    viewOrigin (._ ⊢> (A /> B)) (r⊕⊘ f)    = go (_ ⊢> ((A /> B) <⊕ _)) f  (r⊕⊘ [])
    viewOrigin (._ ⊢> (A /> B)) (d⦸/ f)    = go ((_ ⊗> B) <⊢ _) f            (d⦸/ [])
    viewOrigin (._ ⊢> (A /> B)) (d⊘/ f)    = go ((_ ⊗> B) <⊢ _) f            (d⊘/ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r\⊗ f)    = go (_ ⊢> (_ \> (A <⊕ B))) f  (r\⊗ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r/⊗ f)    = go (_ ⊢> ((A <⊕ B) </ _)) f  (r/⊗ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (m⊕ f₁ f₂) = go (_ ⊢> A)          f₁ (m⊕ᴸ [] f₂)
    viewOrigin (._ ⊢> (A <⊕ B)) (r⦸⊕ f)    = go ((A <⦸ _) <⊢ _) f            (r⦸⊕ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⦸ f)    = go (_ ⊢> (_ ⊕> (A <⊕ B))) f (r⊕⦸ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⊘ f)    = go (_ ⊢> ((A <⊕ B) <⊕ _)) f (r⊕⊘ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r⊘⊕ f)    = go (_ ⊢> A)          f           (r⊘⊕ [])
    viewOrigin (._ ⊢> (A <\ B)) (m\  f₁ f₂) = go (A <⊢ _)         f₁ (m\ᴸ [] f₂)
    viewOrigin (._ ⊢> (A <\ B)) (r\⊗ f)    = go (_ ⊢> (_ \> (A <\ B))) f  (r\⊗ [])
    viewOrigin (._ ⊢> (A <\ B)) (r⊗\ f)    = go ((A <⊗ _) <⊢ _) f            (r⊗\ [])
    viewOrigin (._ ⊢> (A <\ B)) (r/⊗ f)    = go (_ ⊢> ((A <\ B) </ _)) f  (r/⊗ [])
    viewOrigin (._ ⊢> (A <\ B)) (r⊕⦸ f)    = go (_ ⊢> (_ ⊕> (A <\ B))) f  (r⊕⦸ [])
    viewOrigin (._ ⊢> (A <\ B)) (r⊕⊘ f)    = go (_ ⊢> ((A <\ B) <⊕ _)) f  (r⊕⊘ [])
    viewOrigin (._ ⊢> (A <\ B)) (d⦸\ f)    = go ((A <⊗ _) <⊢ _) f            (d⦸\ [])
    viewOrigin (._ ⊢> (A <\ B)) (d⊘\ f)    = go ((A <⊗ _) <⊢ _) f            (d⊘\ [])
    viewOrigin (._ ⊢> (A </ B)) (m/  f₁ f₂) = go (_ ⊢> A)         f₁ (m/ᴸ [] f₂)
    viewOrigin (._ ⊢> (A </ B)) (r\⊗ f)    = go (_ ⊢> (_ \> (A </ B))) f  (r\⊗ [])
    viewOrigin (._ ⊢> (A </ B)) (r/⊗ f)    = go (_ ⊢> ((A </ B) </ _)) f  (r/⊗ [])
    viewOrigin (._ ⊢> (A </ B)) (r⊗/ f)    = go (_ ⊢> A)          f           (r⊗/ [])
    viewOrigin (._ ⊢> (A </ B)) (r⊕⦸ f)    = go (_ ⊢> (_ ⊕> (A </ B))) f  (r⊕⦸ [])
    viewOrigin (._ ⊢> (A </ B)) (r⊕⊘ f)    = go (_ ⊢> ((A </ B) <⊕ _)) f  (r⊕⊘ [])
    viewOrigin (._ ⊢> (A </ B)) (d⦸/ f)    = go (_ ⊢> (_ ⊕> A)) f            (d⦸/ [])
    viewOrigin (._ ⊢> (A </ B)) (d⊘/ f)    = go (_ ⊢> (A <⊕ _)) f            (d⊘/ [])
    private
      go : (I⁺ : Polarisedᴶ + I) (f : LG I [B \ C] ᴶ)
         → (g : LG I [G] ᴶ ⋯ J [G] ᴶ)
         → Origin J⁺ (g $ f)
      go I⁺ f g with viewOrigin I⁺ f
      ... | origin h₁ h₂ f′ pr rewrite pr | ∘-def f′ g (m\ h₁ h₂) = origin h₁ h₂ (g ∘ f′) refl


module / where


data Origin (J⁺ : Polarisedᴶ + J) (f : LG J [B / C] ᴶ) : Set ℓ where
  origin : (h₁ : LG B ⊢ E) (h₂ : LG F ⊢ C)
    → (f′ : LG G ⊢ E / F ⋯ J [G] ᴶ)
    → (pr : f ≡ f′ $ m/ h₁ h₂)
    → Origin J⁺ f


mutual
  viewOrigin : (J⁺ : Polarisedᴶ + J) (f : LG J [B / C] ᴶ) → Origin J⁺ f
  viewOrigin ([] <⊢ ._) (m/          f₁ f₂) = origin f₁ f₂ [] refl
  viewOrigin ([] <⊢ ._) (r⊗\ f)            = go ((_ ⊗> []) <⊢ _) f            (r⊗\ [])
```

```
viewOrigin ([] <⊢ ._) (r⊗/ f)             = go (([] <⊗ _) <⊢ _) f            (r⊗/ [])
viewOrigin ([] <⊢ ._) (r⦸⊕ f)             = go ((_ ⦸> []) <⊢ _) f            (r⦸⊕ [])
viewOrigin ([] <⊢ ._) (r⊘⊕ f)             = go (([] <⊘ _) <⊢ _) f            (r⊘⊕ [])
viewOrigin ((A ⊗> B) <⊢ ._) (m⊗ f_1 f_2) = go (B <⊢ _)         f_2 (m⊗^R f_1 [])
viewOrigin ((A ⊗> B) <⊢ ._) (r\⊗ f)   = go (B <⊢ _)           f          (r\⊗ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A ⊗> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r/⊗ f)   = go (_ ⊢> (_ /> B))  f        (r/⊗ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⊗/ f)   = go (((A ⊗> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A ⊗> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⊘⊕ f)   = go (((A ⊗> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A ⦸> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊗/ f)   = go (((A ⦸> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⦸> B) <⊢ ._) (m⦸ f_1 f_2) = go (B <⊢ _)         f_2 (m⦸^R f_1 [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A ⦸> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊕⦸ f)   = go (B <⊢ _)           f          (r⊕⦸ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊘⊕ f)   = go (((A ⦸> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (d⦸/ f)   = go ((B <⊗ _) <⊢ _) f          (d⦸/ [])
viewOrigin ((A ⦸> B) <⊢ ._) (d⦸\ f)   = go ((_ ⊗> B) <⊢ _) f          (d⦸\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A ⊘> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗/ f)   = go (((A ⊘> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⊘> B) <⊢ ._) (m⊘ f_1 f_2) = go (_ ⊢> B)         f_2 (m⊘^R f_1 [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A ⊘> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊕⊘ f)   = go (_ ⊢> (_ ⊕> B)) f         (r⊕⊘ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊘⊕ f)   = go (((A ⊘> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘\ f)   = go (_ ⊢> (_ ⊕> B)) f         (d⊘\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘/ f)   = go (_ ⊢> (_ ⊕> B)) f         (d⊘/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (m⊗ f_1 f_2) = go (A <⊢ _)         f_1 (m⊗^L [] f_2)
viewOrigin ((A <⊗ B) <⊢ ._) (r\⊗ f)   = go (_ ⊢> (A <\ _))  f        (r\⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A <⊗ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r/⊗ f)   = go (A <⊢ _)           f          (r/⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗/ f)   = go (((A <⊗ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A <⊗ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊘⊕ f)   = go (((A <⊗ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A <⦸ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗/ f)   = go (((A <⦸ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (m⦸ f_1 f_2) = go (_ ⊢> A)         f_1 (m⦸^L [] f_2)
viewOrigin ((A <⦸ B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A <⦸ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊕⦸ f)   = go (_ ⊢> (A <⊕ _)) f         (r⊕⦸ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊘⊕ f)   = go (((A <⦸ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸/ f)   = go (_ ⊢> (A <⊕ _)) f         (d⦸/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸\ f)   = go (_ ⊢> (A <⊕ _)) f         (d⦸\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A <⊘ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗/ f)   = go (((A <⊘ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊘ B) <⊢ ._) (m⊘ f_1 f_2) = go (A <⊢ _)         f_1 (m⊘^L [] f_2)
viewOrigin ((A <⊘ B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A <⊘ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊕⊘ f)   = go (A <⊢ _)           f          (r⊕⊘ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊘⊕ f)   = go (((A <⊘ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘\ f)   = go ((_ ⊗> A) <⊢ _) f          (d⊘\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘/ f)   = go ((A <⊗ _) <⊢ _) f          (d⊘/ [])
```

```
viewOrigin (._ ⊢> (A ⊕> B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A ⊕> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r/⊗ f)   = go (_ ⊢> ((A ⊕> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (m⊕ f1 f2) = go (_ ⊢> B)         f2 (m⊕^R f1 [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⦸⊕ f)  = go (_ ⊢> B)            f        (r⦸⊕ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⦸ f)  = go (_ ⊢> (_ ⊕> (A ⊕> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⊘ f)  = go (_ ⊢> ((A ⊕> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊘⊕ f)  = go ((_ ⊘> B) <⊢ _) f         (r⊘⊕ [])
viewOrigin (._ ⊢> (A \> B)) (m\  f1 f2) = go (_ ⊢> B)        f2 (m\^R f1 [])
viewOrigin (._ ⊢> (A \> B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A \> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊗\ f)   = go (_ ⊢> B)            f        (r⊗\ [])
viewOrigin (._ ⊢> (A \> B)) (r/⊗ f)   = go (_ ⊢> ((A \> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⦸ f)  = go (_ ⊢> (_ ⊕> (A \> B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⊘ f)  = go (_ ⊢> ((A \> B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A \> B)) (d⦸\ f)  = go (_ ⊢> (_ ⊕> B)) f          (d⦸\ [])
viewOrigin (._ ⊢> (A \> B)) (d⊘\ f)  = go (_ ⊢> (B <⊕ _)) f          (d⊘\ [])
viewOrigin (._ ⊢> (A /> B)) (m/  f1 f2) = go (B <⊢ _)        f2 (m/^R f1 [])
viewOrigin (._ ⊢> (A /> B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A /> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r/⊗ f)   = go (_ ⊢> ((A /> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r⊗/ f)   = go ((_ ⊗> B) <⊢ _) f         (r⊗/ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⦸ f)  = go (_ ⊢> (_ ⊕> (A /> B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⊘ f)  = go (_ ⊢> ((A /> B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A /> B)) (d⦸/ f)  = go ((_ ⊗> B) <⊢ _) f         (d⦸/ [])
viewOrigin (._ ⊢> (A /> B)) (d⊘/ f)  = go ((_ ⊗> B) <⊢ _) f         (d⊘/ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A <⊕ B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r/⊗ f)   = go (_ ⊢> ((A <⊕ B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (m⊕ f1 f2) = go (_ ⊢> A)         f1 (m⊕^L [] f2)
viewOrigin (._ ⊢> (A <⊕ B)) (r⦸⊕ f)  = go ((A <⦸ _) <⊢ _) f         (r⦸⊕ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⦸ f)  = go (_ ⊢> (_ ⊕> (A <⊕ B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⊘ f)  = go (_ ⊢> ((A <⊕ B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊘⊕ f)  = go (_ ⊢> A)            f        (r⊘⊕ [])
viewOrigin (._ ⊢> (A <\ B)) (m\  f1 f2) = go (A <⊢ _)        f1 (m\^L [] f2)
viewOrigin (._ ⊢> (A <\ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A <\ B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊗\ f)   = go ((A <⊗ _) <⊢ _) f         (r⊗\ [])
viewOrigin (._ ⊢> (A <\ B)) (r/⊗ f)   = go (_ ⊢> ((A <\ B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊕⦸ f)  = go (_ ⊢> (_ ⊕> (A <\ B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊕⊘ f)  = go (_ ⊢> ((A <\ B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A <\ B)) (d⦸\ f)  = go ((A <⊗ _) <⊢ _) f         (d⦸\ [])
viewOrigin (._ ⊢> (A <\ B)) (d⊘\ f)  = go ((A <⊗ _) <⊢ _) f         (d⊘\ [])
viewOrigin (._ ⊢> (A </ B)) (m/  f1 f2) = go (_ ⊢> A)        f1 (m/^L [] f2)
viewOrigin (._ ⊢> (A </ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A </ B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A </ B)) (r/⊗ f)   = go (_ ⊢> ((A </ B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A </ B)) (r⊗/ f)   = go (_ ⊢> A)            f        (r⊗/ [])
viewOrigin (._ ⊢> (A </ B)) (r⊕⦸ f)  = go (_ ⊢> (_ ⊕> (A </ B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A </ B)) (r⊕⊘ f)  = go (_ ⊢> ((A </ B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A </ B)) (d⦸/ f)  = go (_ ⊢> (_ ⊕> A)) f          (d⦸/ [])
viewOrigin (._ ⊢> (A </ B)) (d⊘/ f)  = go (_ ⊢> (A <⊕ _)) f          (d⊘/ [])
private
  go : (I^+ : Polarised^J + I) (f : LG I [B / C] ^J)
```

```
      → (g : LG I [G] ᴶ ⋯ J [G] ᴶ)
      → Origin J⁺ (g $ f)
    go I⁺ f g with viewOrigin I⁺ f
    ... | origin h₁ h₂ f′ pr rewrite pr | ∘-def f′ g (m/ h₁ h₂) = origin h₁ h₂ (g ∘ f′) refl
```

```
module ⊕ where
```

```
data Origin (J⁺ : Polarisedᴶ + J) (f : LG J [B ⊕ C] ᴶ) : Set ℓ where
  origin : (h₁ : LG B ⊢ E) (h₂ : LG C ⊢ F)
    → (f′ : LG G ⊢ E ⊕ F ⋯ J [G] ᴶ)
    → (pr : f ≡ f′ $ m⊕ h₁ h₂)
    → Origin J⁺ f
```

```
mutual
  viewOrigin : (J⁺ : Polarisedᴶ + J) (f : LG J [B ⊕ C] ᴶ) → Origin J⁺ f
  viewOrigin ([] <⊢ ._) (m⊕       f₁ f₂) = origin f₁ f₂ [] refl
  viewOrigin ([] <⊢ ._) (r⊗\ f)          = go ((_ ⊗> []) <⊢ _) f       (r⊗\ [])
  viewOrigin ([] <⊢ ._) (r⊗/ f)          = go (([] <⊗ _) <⊢ _) f       (r⊗/ [])
  viewOrigin ([] <⊢ ._) (r⦸⊕ f)          = go ((_ ⦸> []) <⊢ _) f       (r⦸⊕ [])
  viewOrigin ([] <⊢ ._) (r⊘⊕ f)          = go (([] <⊘ _) <⊢ _) f       (r⊘⊕ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (m⊗ f₁ f₂) = go (B <⊢ _)       f₂ (m⊗ᴿ f₁ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r\⊗ f)  = go (B <⊢ _)         f       (r\⊗ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A ⊗> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r/⊗ f)  = go (_ ⊢> (_ /> B))  f       (r/⊗ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊗/ f)  = go (((A ⊗> B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⦸⊕ f)  = go ((_ ⦸> (A ⊗> B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊘⊕ f)  = go (((A ⊗> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A ⦸> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊗/ f)  = go (((A ⦸> B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (m⦸ f₁ f₂) = go (B <⊢ _)       f₂ (m⦸ᴿ f₁ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⦸⊕ f)  = go ((_ ⦸> (A ⦸> B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊕⦸ f)  = go (B <⊢ _)         f       (r⊕⦸ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊘⊕ f)  = go (((A ⦸> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (d⦸/ f)  = go ((B <⊗ _) <⊢ _) f        (d⦸/ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (d⦸\ f)  = go ((_ ⊗> B) <⊢ _) f        (d⦸\ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A ⊘> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⊗/ f)  = go (((A ⊘> B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (m⊘ f₁ f₂) = go (_ ⊢> B)       f₂ (m⊘ᴿ f₁ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⦸⊕ f)  = go ((_ ⦸> (A ⊘> B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⊕⊘ f)  = go (_ ⊢> (_ ⊕> B)) f        (r⊕⊘ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (r⊘⊕ f)  = go (((A ⊘> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (d⊘\ f)  = go (_ ⊢> (_ ⊕> B)) f        (d⊘\ [])
  viewOrigin ((A ⊘> B) <⊢ ._) (d⊘/ f)  = go (_ ⊢> (_ ⊕> B)) f        (d⊘/ [])
  viewOrigin ((A <⊗ B) <⊢ ._) (m⊗ f₁ f₂) = go (A <⊢ _)       f₁ (m⊗ᴸ [] f₂)
  viewOrigin ((A <⊗ B) <⊢ ._) (r\⊗ f)  = go (_ ⊢> (A <\ _))  f       (r\⊗ [])
  viewOrigin ((A <⊗ B) <⊢ ._) (r⊗\ f)  = go ((_ ⊗> (A <⊗ B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A <⊗ B) <⊢ ._) (r/⊗ f)  = go (A <⊢ _)         f       (r/⊗ [])
```

```
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗/ f)    = go (((A <⊗ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⦸⊕ f)    = go ((_ ⦸> (A <⊗ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊘⊕ f)    = go (((A <⊗ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗\ f)    = go ((_ ⊗> (A <⦸ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗/ f)    = go (((A <⦸ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (m⦸ f1 f2) = go (_ ⊢> A)        f1 (m⦸^L [] f2)
viewOrigin ((A <⦸ B) <⊢ ._) (r⦸⊕ f)    = go ((_ ⦸> (A <⦸ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊕⦸ f)    = go (_ ⊢> (A <⊕ _)) f         (r⊕⦸ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊘⊕ f)    = go (((A <⦸ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸/ f)    = go (_ ⊢> (A <⊕ _)) f         (d⦸/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸\ f)    = go (_ ⊢> (A <⊕ _)) f         (d⦸\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗\ f)    = go ((_ ⊗> (A <⊘ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗/ f)    = go (((A <⊘ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊘ B) <⊢ ._) (m⊘ f1 f2) = go (A <⊢ _)        f1 (m⊘^L [] f2)
viewOrigin ((A <⊘ B) <⊢ ._) (r⦸⊕ f)    = go ((_ ⦸> (A <⊘ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊕⊘ f)    = go (A <⊢ _)           f      (r⊕⊘ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊘⊕ f)    = go (((A <⊘ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘\ f)    = go ((_ ⊗> A) <⊢ _) f         (d⊘\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘/ f)    = go ((A <⊗ _) <⊢ _) f         (d⊘/ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r\⊗ f)    = go (_ ⊢> (_ \> (A ⊕> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r/⊗ f)    = go (_ ⊢> ((A ⊕> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (m⊕ f1 f2) = go (_ ⊢> B)        f2 (m⊕^R f1 [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⦸⊕ f)    = go (_ ⊢> B)           f      (r⦸⊕ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⦸ f)    = go (_ ⊢> (_ ⊕> (A ⊕> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⊘ f)    = go (_ ⊢> ((A ⊕> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊘⊕ f)    = go ((_ ⊘> B) <⊢ _) f         (r⊘⊕ [])
viewOrigin (._ ⊢> (A \> B)) (m\  f1 f2) = go (_ ⊢> B)        f2 (m\^R f1 [])
viewOrigin (._ ⊢> (A \> B)) (r\⊗ f)     = go (_ ⊢> (_ \> (A \> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊗\ f)     = go (_ ⊢> B)           f      (r⊗\ [])
viewOrigin (._ ⊢> (A \> B)) (r/⊗ f)     = go (_ ⊢> ((A \> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⦸ f)     = go (_ ⊢> (_ ⊕> (A \> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⊘ f)     = go (_ ⊢> ((A \> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A \> B)) (d⦸\ f)     = go (_ ⊢> (_ ⊕> B)) f         (d⦸\ [])
viewOrigin (._ ⊢> (A \> B)) (d⊘\ f)     = go (_ ⊢> (B <⊕ _)) f         (d⊘\ [])
viewOrigin (._ ⊢> (A /> B)) (m/  f1 f2) = go (B <⊢ _)        f2 (m/^R f1 [])
viewOrigin (._ ⊢> (A /> B)) (r\⊗ f)     = go (_ ⊢> (_ \> (A /> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r/⊗ f)     = go (_ ⊢> ((A /> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r⊗/ f)     = go ((_ ⊗> B) <⊢ _) f         (r⊗/ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⦸ f)     = go (_ ⊢> (_ ⊕> (A /> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⊘ f)     = go (_ ⊢> ((A /> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A /> B)) (d⦸/ f)     = go ((_ ⊗> B) <⊢ _) f         (d⦸/ [])
viewOrigin (._ ⊢> (A /> B)) (d⊘/ f)     = go ((_ ⊗> B) <⊢ _) f         (d⊘/ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r\⊗ f)    = go (_ ⊢> (_ \> (A <⊕ B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r/⊗ f)    = go (_ ⊢> ((A <⊕ B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (m⊕ f1 f2) = go (_ ⊢> A)        f1 (m⊕^L [] f2)
viewOrigin (._ ⊢> (A <⊕ B)) (r⦸⊕ f)    = go ((A <⦸ _) <⊢ _) f         (r⦸⊕ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⦸ f)    = go (_ ⊢> (_ ⊕> (A <⊕ B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⊘ f)    = go (_ ⊢> ((A <⊕ B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊘⊕ f)    = go (_ ⊢> A)           f      (r⊘⊕ [])
```

```
  viewOrigin (._ ⊢> (A <\ B)) (m\  f₁ f₂) = go (A <⊢ _)       f₁ (m\ᴸ [] f₂)
  viewOrigin (._ ⊢> (A <\ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A <\ B))) f  (r\⊗ [])
  viewOrigin (._ ⊢> (A <\ B)) (r⊗\ f)   = go ((A <⊗ _) <⊢ _) f        (r⊗\ [])
  viewOrigin (._ ⊢> (A <\ B)) (r/⊗ f)   = go (_ ⊢> ((A <\ B) </ _)) f  (r/⊗ [])
  viewOrigin (._ ⊢> (A <\ B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A <\ B))) f  (r⊕⦸ [])
  viewOrigin (._ ⊢> (A <\ B)) (r⊕⊘ f)   = go (_ ⊢> ((A <\ B) <⊕ _)) f  (r⊕⊘ [])
  viewOrigin (._ ⊢> (A <\ B)) (d⦸\ f)   = go ((A <⊗ _) <⊢ _) f        (d⦸\ [])
  viewOrigin (._ ⊢> (A <\ B)) (d⊘\ f)   = go ((A <⊗ _) <⊢ _) f        (d⊘\ [])
  viewOrigin (._ ⊢> (A </ B)) (m/  f₁ f₂) = go (_ ⊢> A)       f₁ (m/ᴸ [] f₂)
  viewOrigin (._ ⊢> (A </ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A </ B))) f  (r\⊗ [])
  viewOrigin (._ ⊢> (A </ B)) (r/⊗ f)   = go (_ ⊢> ((A </ B) </ _)) f  (r/⊗ [])
  viewOrigin (._ ⊢> (A </ B)) (r⊗/ f)   = go (_ ⊢> A)          f        (r⊗/ [])
  viewOrigin (._ ⊢> (A </ B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A </ B))) f  (r⊕⦸ [])
  viewOrigin (._ ⊢> (A </ B)) (r⊕⊘ f)   = go (_ ⊢> ((A </ B) <⊕ _)) f  (r⊕⊘ [])
  viewOrigin (._ ⊢> (A </ B)) (d⦸/ f)   = go (_ ⊢> (_ ⊕> A)) f        (d⦸/ [])
  viewOrigin (._ ⊢> (A </ B)) (d⊘/ f)   = go (_ ⊢> (A <⊕ _)) f        (d⊘/ [])
  private
    go : (I⁺ : Polarisedᴶ + I) (f : LG I [B ⊕ C] ᴶ)
      → (g : LG I [G] ᴶ ⋯ J [G] ᴶ)
      → Origin J⁺ (g $ f)
    go I⁺ f g with viewOrigin I⁺ f
    ... | origin h₁ h₂ f′ pr rewrite pr | ∘-def f′ g (m⊕ h₁ h₂) = origin h₁ h₂ (g ∘ f′) refl


module ⊘ where


data Origin (J⁻ : Polarisedᴶ $-$ J) (f : LG J [B ⊘ C] ᴶ) : Set ℓ where
  origin : (h₁ : LG E ⊢ B) (h₂ : LG C ⊢ F)
    → (f′ : LG E ⊘ F ⊢ G ⋯ J [G] ᴶ)
    → (pr : f ≡ f′ $ m⊘ h₁ h₂)
    → Origin J⁻ f


mutual
  viewOrigin : (J⁻ : Polarisedᴶ $-$ J) (f : LG J [B ⊘ C] ᴶ) → Origin J⁻ f
  viewOrigin (._ ⊢> []) (m⊘          f₁ f₂) = origin f₁ f₂ [] refl
  viewOrigin (._ ⊢> []) (r\⊗ f)           = go (_ ⊢> (_ \> [])) f         (r\⊗ [])
  viewOrigin (._ ⊢> []) (r/⊗ f)           = go (_ ⊢> ([] </ _)) f         (r/⊗ [])
  viewOrigin (._ ⊢> []) (r⊕⦸ f)           = go (_ ⊢> (_ ⊕> [])) f         (r⊕⦸ [])
  viewOrigin (._ ⊢> []) (r⊕⊘ f)           = go (_ ⊢> ([] <⊕ _)) f         (r⊕⊘ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (m⊗ f₁ f₂) = go (B <⊢ _)       f₂ (m⊗ᴿ f₁ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r\⊗ f)   = go (B <⊢ _)          f          (r\⊗ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A ⊗> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r/⊗ f)   = go (_ ⊢> (_ /> B)) f          (r/⊗ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊗/ f)   = go (((A ⊗> B) <⊗ _) <⊢ _) f (r⊗/ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⦸⊕ f)   = go ((_ ⦸> (A ⊗> B)) <⊢ _) f (r⦸⊕ [])
  viewOrigin ((A ⊗> B) <⊢ ._) (r⊘⊕ f)   = go (((A ⊗> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊗\ f)   = go ((_ ⊗> (A ⦸> B)) <⊢ _) f (r⊗\ [])
  viewOrigin ((A ⦸> B) <⊢ ._) (r⊗/ f)   = go (((A ⦸> B) <⊗ _) <⊢ _) f (r⊗/ [])
```

```
viewOrigin ((A ⦸> B) <⊢ ._) (m⦸ f1 f2)  =  go (B <⊢ _)        f2 (m⦸^R f1 [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⦸⊕ f)  =  go ((_ ⦸> (A ⦸> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊕⦸ f)  =  go (B <⊢ _)           f        (r⊕⦸ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊘⊕ f)  =  go (((A ⦸> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (d⦸/ f)  =  go ((B <⊗ _) <⊢ _) f       (d⦸/ [])
viewOrigin ((A ⦸> B) <⊢ ._) (d⦸\ f)  =  go ((_ ⊗> B) <⊢ _) f       (d⦸\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A ⊘> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗/ f)   =  go (((A ⊘> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⊘> B) <⊢ ._) (m⊘ f1 f2)  =  go (_ ⊢> B)        f2 (m⊘^R f1 [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⦸⊕ f)  =  go ((_ ⦸> (A ⊘> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊕⊘ f)  =  go (_ ⊢> (_ ⊕> B)) f       (r⊕⊘ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊘⊕ f)  =  go (((A ⊘> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘\ f)  =  go (_ ⊢> (_ ⊕> B)) f       (d⊘\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘/ f)  =  go (_ ⊢> (_ ⊕> B)) f       (d⊘/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (m⊗ f1 f2)  =  go (A <⊢ _)        f1 (m⊗^L [] f2)
viewOrigin ((A <⊗ B) <⊢ ._) (r\⊗ f)   =  go (_ ⊢> (A <\ _))  f       (r\⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A <⊗ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r/⊗ f)   =  go (A <⊢ _)           f        (r/⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗/ f)   =  go (((A <⊗ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⦸⊕ f)  =  go ((_ ⦸> (A <⊗ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊘⊕ f)  =  go (((A <⊗ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A <⦸ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗/ f)   =  go (((A <⦸ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (m⦸ f1 f2)  =  go (_ ⊢> A)        f1 (m⦸^L [] f2)
viewOrigin ((A <⦸ B) <⊢ ._) (r⦸⊕ f)  =  go ((_ ⦸> (A <⦸ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊕⦸ f)  =  go (_ ⊢> (A <⊕ _)) f       (r⊕⦸ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊘⊕ f)  =  go (((A <⦸ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸/ f)  =  go (_ ⊢> (A <⊕ _)) f       (d⦸/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸\ f)  =  go (_ ⊢> (A <⊕ _)) f       (d⦸\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗\ f)   =  go ((_ ⊗> (A <⊘ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗/ f)   =  go (((A <⊘ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊘ B) <⊢ ._) (m⊘ f1 f2)  =  go (A <⊢ _)        f1 (m⊘^L [] f2)
viewOrigin ((A <⊘ B) <⊢ ._) (r⦸⊕ f)  =  go ((_ ⦸> (A <⊘ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊕⊘ f)  =  go (A <⊢ _)           f        (r⊕⊘ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊘⊕ f)  =  go (((A <⊘ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘\ f)  =  go ((_ ⊗> A) <⊢ _) f       (d⊘\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘/ f)  =  go ((A <⊗ _) <⊢ _) f       (d⊘/ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r\⊗ f)   =  go (_ ⊢> (_ \> (A ⊕> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r/⊗ f)   =  go (_ ⊢> ((A ⊕> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (m⊕ f1 f2)  =  go (_ ⊢> B)        f2 (m⊕^R f1 [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⦸⊕ f)  =  go (_ ⊢> B)           f        (r⦸⊕ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⦸ f)  =  go (_ ⊢> (_ ⊕> (A ⊕> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⊘ f)  =  go (_ ⊢> ((A ⊕> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊘⊕ f)  =  go ((_ ⊘> B) <⊢ _) f       (r⊘⊕ [])
viewOrigin (._ ⊢> (A \> B)) (m\  f1 f2)  =  go (_ ⊢> B)        f2 (m\^R f1 [])
viewOrigin (._ ⊢> (A \> B)) (r\⊗ f)    =  go (_ ⊢> (_ \> (A \> B))) f   (r\⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊗\ f)    =  go (_ ⊢> B)           f        (r⊗\ [])
viewOrigin (._ ⊢> (A \> B)) (r/⊗ f)    =  go (_ ⊢> ((A \> B) </ _)) f   (r/⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⦸ f)    =  go (_ ⊢> (_ ⊕> (A \> B))) f  (r⊕⦸ [])
```

```
    viewOrigin (._ ⊢> (A \> B)) (r⊕⊘ f)   = go (_ ⊢> ((A \> B) <⊕ _)) f  (r⊕⊘ [])
    viewOrigin (._ ⊢> (A \> B)) (d⦸\ f)   = go (_ ⊢> (_ ⊕> B)) f          (d⦸\ [])
    viewOrigin (._ ⊢> (A \> B)) (d⊘\ f)   = go (_ ⊢> (B <⊕ _)) f          (d⊘\ [])
    viewOrigin (._ ⊢> (A /> B)) (m/  f₁ f₂) = go (B <⊢ _)      f₂ (m/ᴿ f₁ [])
    viewOrigin (._ ⊢> (A /> B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A /> B))) f  (r\⊗ [])
    viewOrigin (._ ⊢> (A /> B)) (r/⊗ f)   = go (_ ⊢> ((A /> B) </ _)) f  (r/⊗ [])
    viewOrigin (._ ⊢> (A /> B)) (r⊗/ f)   = go ((_ ⊗> B) <⊢ _) f          (r⊗/ [])
    viewOrigin (._ ⊢> (A /> B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A /> B))) f  (r⊕⦸ [])
    viewOrigin (._ ⊢> (A /> B)) (r⊕⊘ f)   = go (_ ⊢> ((A /> B) <⊕ _)) f  (r⊕⊘ [])
    viewOrigin (._ ⊢> (A /> B)) (d⦸/ f)   = go ((_ ⊗> B) <⊢ _) f          (d⦸/ [])
    viewOrigin (._ ⊢> (A /> B)) (d⊘/ f)   = go ((_ ⊗> B) <⊢ _) f          (d⊘/ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A <⊕ B))) f  (r\⊗ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r/⊗ f)   = go (_ ⊢> ((A <⊕ B) </ _)) f  (r/⊗ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (m⊕ f₁ f₂) = go (_ ⊢> A)       f₁ (m⊕ᴸ [] f₂)
    viewOrigin (._ ⊢> (A <⊕ B)) (r⦸⊕ f)   = go ((A <⦸ _) <⊢ _) f          (r⦸⊕ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A <⊕ B))) f (r⊕⦸ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⊘ f)   = go (_ ⊢> ((A <⊕ B) <⊕ _)) f (r⊕⊘ [])
    viewOrigin (._ ⊢> (A <⊕ B)) (r⊘⊕ f)   = go (_ ⊢> A)         f          (r⊘⊕ [])
    viewOrigin (._ ⊢> (A <\ B)) (m\  f₁ f₂) = go (A <⊢ _)      f₁ (m\ᴸ [] f₂)
    viewOrigin (._ ⊢> (A <\ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A <\ B))) f  (r\⊗ [])
    viewOrigin (._ ⊢> (A <\ B)) (r⊗\ f)   = go ((A <⊗ _) <⊢ _) f          (r⊗\ [])
    viewOrigin (._ ⊢> (A <\ B)) (r/⊗ f)   = go (_ ⊢> ((A <\ B) </ _)) f  (r/⊗ [])
    viewOrigin (._ ⊢> (A <\ B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A <\ B))) f  (r⊕⦸ [])
    viewOrigin (._ ⊢> (A <\ B)) (r⊕⊘ f)   = go (_ ⊢> ((A <\ B) <⊕ _)) f  (r⊕⊘ [])
    viewOrigin (._ ⊢> (A <\ B)) (d⦸\ f)   = go ((A <⊗ _) <⊢ _) f          (d⦸\ [])
    viewOrigin (._ ⊢> (A <\ B)) (d⊘\ f)   = go ((A <⊗ _) <⊢ _) f          (d⊘\ [])
    viewOrigin (._ ⊢> (A </ B)) (m/  f₁ f₂) = go (_ ⊢> A)      f₁ (m/ᴸ [] f₂)
    viewOrigin (._ ⊢> (A </ B)) (r\⊗ f)   = go (_ ⊢> (_ \> (A </ B))) f  (r\⊗ [])
    viewOrigin (._ ⊢> (A </ B)) (r/⊗ f)   = go (_ ⊢> ((A </ B) </ _)) f  (r/⊗ [])
    viewOrigin (._ ⊢> (A </ B)) (r⊗/ f)   = go (_ ⊢> A)         f          (r⊗/ [])
    viewOrigin (._ ⊢> (A </ B)) (r⊕⦸ f)   = go (_ ⊢> (_ ⊕> (A </ B))) f  (r⊕⦸ [])
    viewOrigin (._ ⊢> (A </ B)) (r⊕⊘ f)   = go (_ ⊢> ((A </ B) <⊕ _)) f  (r⊕⊘ [])
    viewOrigin (._ ⊢> (A </ B)) (d⦸/ f)   = go (_ ⊢> (_ ⊕> A)) f          (d⦸/ [])
    viewOrigin (._ ⊢> (A </ B)) (d⊘/ f)   = go (_ ⊢> (A <⊕ _)) f          (d⊘/ [])
    private
      go : (I⁻ : Polarisedᴶ $-$ I) (f : LG I [B ⊘ C] ᴶ)
         → (g : LG I [G] ᴶ ⋯ J [G] ᴶ)
         → Origin J⁻ (g $ f)
      go I⁻ f g with viewOrigin I⁻ f
      ... | origin h₁ h₂ f′ pr rewrite pr | ∘-def f′ g (m⊘ h₁ h₂) = origin h₁ h₂ (g ∘ f′) refl


module ⦸ where


data Origin (J⁻ : Polarisedᴶ $-$ J) (f : LG J [B ⦸ C] ᴶ) : Set ℓ where
  origin : (h₁ : LG B ⊢ E) (h₂ : LG F ⊢ C)
    → (f′ : LG E ⦸ F ⊢ G ⋯ J [G] ᴶ)
    → (pr : f ≡ f′ $ m⦸ h₁ h₂)
    → Origin J⁻ f
```

mutual

viewOrigin : (J⁻ : Polarised^J $-$ J) (f : LG J [B ⦸ C] ^{J}) → Origin J⁻ f
viewOrigin (._ ⊢> []) (m⦸ f_1 f_2) = origin f_1 f_2 [] refl
viewOrigin (._ ⊢> []) (r\⊗ f) = go (_ ⊢> (_ \> [])) f (r\⊗ [])
viewOrigin (._ ⊢> []) (r/⊗ f) = go (_ ⊢> ([] </ _)) f (r/⊗ [])
viewOrigin (._ ⊢> []) (r⊕⦸ f) = go (_ ⊢> (_ ⊕> [])) f (r⊕⦸ [])
viewOrigin (._ ⊢> []) (r⊕⊘ f) = go (_ ⊢> ([] <⊕ _)) f (r⊕⊘ [])
viewOrigin ((A ⊗> B) <⊢ ._) (m⊗ f_1 f_2) = go (B <⊢ _) f_2 (m$\otimes^R$ f_1 [])
viewOrigin ((A ⊗> B) <⊢ ._) (r\⊗ f) = go (B <⊢ _) f (r\⊗ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⊗\ f) = go ((_ ⊗> (A ⊗> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r/⊗ f) = go (_ ⊢> (_ /> B)) f (r/⊗ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⊗/ f) = go (((A ⊗> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⦸⊕ f) = go ((_ ⦸> (A ⊗> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⊗> B) <⊢ ._) (r⊘⊕ f) = go (((A ⊗> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊗\ f) = go ((_ ⊗> (A ⦸> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊗/ f) = go (((A ⦸> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⦸> B) <⊢ ._) (m⦸ f_1 f_2) = go (B <⊢ _) f_2 (m$⦸^R$ f_1 [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⦸⊕ f) = go ((_ ⦸> (A ⦸> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊕⦸ f) = go (B <⊢ _) f (r⊕⦸ [])
viewOrigin ((A ⦸> B) <⊢ ._) (r⊘⊕ f) = go (((A ⦸> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⦸> B) <⊢ ._) (d⦸/ f) = go ((B <⊗ _) <⊢ _) f (d⦸/ [])
viewOrigin ((A ⦸> B) <⊢ ._) (d⦸\ f) = go ((_ ⊗> B) <⊢ _) f (d⦸\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗\ f) = go ((_ ⊗> (A ⊘> B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊗/ f) = go (((A ⊘> B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A ⊘> B) <⊢ ._) (m⊘ f_1 f_2) = go (_ ⊢> B) f_2 (m$\oslash^R$ f_1 [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⦸⊕ f) = go ((_ ⦸> (A ⊘> B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊕⊘ f) = go (_ ⊢> (_ ⊕> B)) f (r⊕⊘ [])
viewOrigin ((A ⊘> B) <⊢ ._) (r⊘⊕ f) = go (((A ⊘> B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘\ f) = go (_ ⊢> (_ ⊕> B)) f (d⊘\ [])
viewOrigin ((A ⊘> B) <⊢ ._) (d⊘/ f) = go (_ ⊢> (_ ⊕> B)) f (d⊘/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (m⊗ f_1 f_2) = go (A <⊢ _) f_1 (m$\otimes^L$ [] f_2)
viewOrigin ((A <⊗ B) <⊢ ._) (r\⊗ f) = go (_ ⊢> (A <\ _)) f (r\⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗\ f) = go ((_ ⊗> (A <⊗ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r/⊗ f) = go (A <⊢ _) f (r/⊗ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊗/ f) = go (((A <⊗ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⦸⊕ f) = go ((_ ⦸> (A <⊗ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⊗ B) <⊢ ._) (r⊘⊕ f) = go (((A <⊗ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗\ f) = go ((_ ⊗> (A <⦸ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊗/ f) = go (((A <⦸ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (m⦸ f_1 f_2) = go (_ ⊢> A) f_1 (m$⦸^L$ [] f_2)
viewOrigin ((A <⦸ B) <⊢ ._) (r⦸⊕ f) = go ((_ ⦸> (A <⦸ B)) <⊢ _) f (r⦸⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊕⦸ f) = go (_ ⊢> (A <⊕ _)) f (r⊕⦸ [])
viewOrigin ((A <⦸ B) <⊢ ._) (r⊘⊕ f) = go (((A <⦸ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸/ f) = go (_ ⊢> (A <⊕ _)) f (d⦸/ [])
viewOrigin ((A <⦸ B) <⊢ ._) (d⦸\ f) = go (_ ⊢> (A <⊕ _)) f (d⦸\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗\ f) = go ((_ ⊗> (A <⊘ B)) <⊢ _) f (r⊗\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊗/ f) = go (((A <⊘ B) <⊗ _) <⊢ _) f (r⊗/ [])
viewOrigin ((A <⊘ B) <⊢ ._) (m⊘ f_1 f_2) = go (A <⊢ _) f_1 (m$\oslash^L$ [] f_2)
viewOrigin ((A <⊘ B) <⊢ ._) (r⦸⊕ f) = go ((_ ⦸> (A <⊘ B)) <⊢ _) f (r⦸⊕ [])

```
viewOrigin ((A <⊘ B) <⊢ ._) (r⊕⊘ f)  =  go (A <⊢ _)            f            (r⊕⊘ [])
viewOrigin ((A <⊘ B) <⊢ ._) (r⊘⊕ f)  =  go (((A <⊘ B) <⊘ _) <⊢ _) f (r⊘⊕ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘\ f)  =  go ((_ ⊗> A) <⊢ _) f           (d⊘\ [])
viewOrigin ((A <⊘ B) <⊢ ._) (d⊘/ f)  =  go ((A <⊗ _) <⊢ _) f           (d⊘/ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r\⊗ f)   =  go (_ ⊢> (_ \> (A ⊕> B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r/⊗ f)   =  go (_ ⊢> ((A ⊕> B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A ⊕> B)) (m⊕ f_1 f_2)  =  go (_ ⊢> B)        f_2 (m⊕^R f_1 [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⦸⊕ f)  =  go (_ ⊢> B)            f            (r⦸⊕ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⦸ f)  =  go (_ ⊢> (_ ⊕> (A ⊕> B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊕⊘ f)  =  go (_ ⊢> ((A ⊕> B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A ⊕> B)) (r⊘⊕ f)  =  go ((_ ⊘> B) <⊢ _) f           (r⊘⊕ [])
viewOrigin (._ ⊢> (A \> B)) (m\  f_1 f_2)  =  go (_ ⊢> B)        f_2 (m\^R f_1 [])
viewOrigin (._ ⊢> (A \> B)) (r\⊗ f)    =  go (_ ⊢> (_ \> (A \> B))) f   (r\⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊗\ f)    =  go (_ ⊢> B)            f            (r⊗\ [])
viewOrigin (._ ⊢> (A \> B)) (r/⊗ f)    =  go (_ ⊢> ((A \> B) </ _)) f   (r/⊗ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⦸ f)   =  go (_ ⊢> (_ ⊕> (A \> B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A \> B)) (r⊕⊘ f)   =  go (_ ⊢> ((A \> B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A \> B)) (d⦸\ f)   =  go (_ ⊢> (_ ⊕> B)) f           (d⦸\ [])
viewOrigin (._ ⊢> (A \> B)) (d⊘\ f)   =  go (_ ⊢> (B <⊕ _)) f           (d⊘\ [])
viewOrigin (._ ⊢> (A /> B)) (m/  f_1 f_2)  =  go (B <⊢ _)        f_2 (m/^R f_1 [])
viewOrigin (._ ⊢> (A /> B)) (r\⊗ f)    =  go (_ ⊢> (_ \> (A /> B))) f   (r\⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r/⊗ f)    =  go (_ ⊢> ((A /> B) </ _)) f   (r/⊗ [])
viewOrigin (._ ⊢> (A /> B)) (r⊗/ f)    =  go ((_ ⊗> B) <⊢ _) f           (r⊗/ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⦸ f)   =  go (_ ⊢> (_ ⊕> (A /> B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A /> B)) (r⊕⊘ f)   =  go (_ ⊢> ((A /> B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A /> B)) (d⦸/ f)   =  go ((_ ⊗> B) <⊢ _) f           (d⦸/ [])
viewOrigin (._ ⊢> (A /> B)) (d⊘/ f)   =  go ((_ ⊗> B) <⊢ _) f           (d⊘/ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r\⊗ f)   =  go (_ ⊢> (_ \> (A <⊕ B))) f  (r\⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r/⊗ f)   =  go (_ ⊢> ((A <⊕ B) </ _)) f  (r/⊗ [])
viewOrigin (._ ⊢> (A <⊕ B)) (m⊕ f_1 f_2)  =  go (_ ⊢> A)        f_1 (m⊕^L [] f_2)
viewOrigin (._ ⊢> (A <⊕ B)) (r⦸⊕ f)  =  go ((A <⦸ _) <⊢ _) f           (r⦸⊕ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⦸ f)  =  go (_ ⊢> (_ ⊕> (A <⊕ B))) f (r⊕⦸ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊕⊘ f)  =  go (_ ⊢> ((A <⊕ B) <⊕ _)) f (r⊕⊘ [])
viewOrigin (._ ⊢> (A <⊕ B)) (r⊘⊕ f)  =  go (_ ⊢> A)            f            (r⊘⊕ [])
viewOrigin (._ ⊢> (A <\ B)) (m\  f_1 f_2)  =  go (A <⊢ _)        f_1 (m\^L [] f_2)
viewOrigin (._ ⊢> (A <\ B)) (r\⊗ f)    =  go (_ ⊢> (_ \> (A <\ B))) f   (r\⊗ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊗\ f)    =  go ((A <⊗ _) <⊢ _) f           (r⊗\ [])
viewOrigin (._ ⊢> (A <\ B)) (r/⊗ f)    =  go (_ ⊢> ((A <\ B) </ _)) f   (r/⊗ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊕⦸ f)   =  go (_ ⊢> (_ ⊕> (A <\ B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A <\ B)) (r⊕⊘ f)   =  go (_ ⊢> ((A <\ B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A <\ B)) (d⦸\ f)   =  go ((A <⊗ _) <⊢ _) f           (d⦸\ [])
viewOrigin (._ ⊢> (A <\ B)) (d⊘\ f)   =  go ((A <⊗ _) <⊢ _) f           (d⊘\ [])
viewOrigin (._ ⊢> (A </ B)) (m/  f_1 f_2)  =  go (_ ⊢> A)        f_1 (m/^L [] f_2)
viewOrigin (._ ⊢> (A </ B)) (r\⊗ f)    =  go (_ ⊢> (_ \> (A </ B))) f   (r\⊗ [])
viewOrigin (._ ⊢> (A </ B)) (r/⊗ f)    =  go (_ ⊢> ((A </ B) </ _)) f   (r/⊗ [])
viewOrigin (._ ⊢> (A </ B)) (r⊗/ f)    =  go (_ ⊢> A)            f            (r⊗/ [])
viewOrigin (._ ⊢> (A </ B)) (r⊕⦸ f)   =  go (_ ⊢> (_ ⊕> (A </ B))) f  (r⊕⦸ [])
viewOrigin (._ ⊢> (A </ B)) (r⊕⊘ f)   =  go (_ ⊢> ((A </ B) <⊕ _)) f  (r⊕⊘ [])
viewOrigin (._ ⊢> (A </ B)) (d⦸/ f)   =  go (_ ⊢> (_ ⊕> A)) f           (d⦸/ [])
```

```
  viewOrigin (._ ⊢> (A </ B)) (d⊘/ f)   = go (_ ⊢> (A <⊕ _)) f            (d⊘/ [])
  private
    go : (I⁻ : Polarised^J $-$ I) (f : LG I [B ⦸ C] ^J)
       → (g : LG I [G] ^J ⋯ J [G] ^J)
       → Origin J⁻ (g $ f)
    go I⁻ f g with viewOrigin I⁻ f
    ... | origin h₁ h₂ f′ pr rewrite pr | ∘-def f′ g (m⦸ h₁ h₂) = origin h₁ h₂ (g ∘ f′) refl

trans′ : (f : LG A ⊢ B) (g : LG B ⊢ C) → LG A ⊢ C
trans′ f g with el.viewOrigin ([] <⊢ _) g
... | el.origin g′ _ = g′ $ f
trans′ f g with ⊗.viewOrigin (_ ⊢> []) f
... | ⊗.origin h₁ h₂ f′ _ = f′ $ r/⊗ (trans′ h₁ (r⊗/ (r\⊗ (trans′ h₂ (r⊗\ g)))))
trans′ f g with /.viewOrigin ([] <⊢ _) g
... | /.origin h₁ h₂ g′ _ = g′ $ r⊗/ (r\⊗ (trans′ h₂ (r⊗\ (trans′ (r/⊗ f) h₁))))
trans′ f g with \.viewOrigin ([] <⊢ _) g
... | \.origin h₁ h₂ g′ _ = g′ $ r⊗\ (r/⊗ (trans′ h₁ (r⊗/ (trans′ (r\⊗ f) h₂))))
trans′ f g with ⊕.viewOrigin ([] <⊢ _) g
... | ⊕.origin h₁ h₂ g′ _ = g′ $ r⊘⊕ (trans′ (r⊕⊘ (r⦸⊕ (trans′ (r⊕⦸ f) h₂))) h₁)
trans′ f g with ⊘.viewOrigin (_ ⊢> []) f
... | ⊘.origin h₁ h₂ f′ _ = f′ $ r⊕⊘ (r⦸⊕ (trans′ (r⊕⦸ (trans′ h₁ (r⊘⊕ g))) h₂))
trans′ f g with ⦸.viewOrigin (_ ⊢> []) f
... | ⦸.origin h₁ h₂ f′ _ = f′ $ r⊕⦸ (r⊘⊕ (trans′ (r⊕⊘ (trans′ h₂ (r⦸⊕ g))) h₁))
```

References

Bastenhof, A. (2013). Categorial symmetry.

Curien, P.-L. and Herbelin, H. (2000). The duality of computation. *SIGPLAN Not.*, 35(9):233–243.

de Groote, P. (1994). A cps-translation of the λμ-calculus.

de Groote, P. (2001). Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pages 148–155.

Érdi, G. (2013). Simply typed lambda calculus in agda, without shortcuts. http://gergo.erdi.hu/blog/2013-05-01-simply_typed_lambda_calculus_in_agda,_without_shortcuts/. Accessed: 2014-01-28.

Girard, J.-Y. (1991). A new constructive logic: classic logic. *Mathematical Structures in Computer Science*, 1:255–296.

Kiselyov, O., Lämmel, R., and Schupke, K. (2004). Strongly typed heterogeneous collections. In *Proceedings of the 2004 ACM SIGPLAN Workshop on Haskell*, Haskell '04, pages 96–107, New York, NY, USA. ACM.

Lindblad, F. and Benke, M. (2006). A tool for automated theorem proving in agda. In *Proceedings of the 2004 International Conference on Types for Proofs and Programs*, TYPES'04, pages 154–169, Berlin, Heidelberg. Springer-Verlag.

Martin-Löf, P. (1984). Intuitionistic type theory.

Mazzoli, F. (2013). Agda by example: λ-calculus. http://mazzo.li/posts/Lambda.html. Accessed: 2014-01-28.

Moortgat, M. (2009). Symmetric categorial grammar. *Journal of Philosophical Logic*, 38(6):681–710.

Moortgat, M. and Moot, R. (2013). Proofs nets and the categorial flow of information.

Moortgat, M. and Oehrle, R. (1999). Proof nets for the grammatical base logic. *Dynamic perspectives in logic and linguistics, Roma, Bulzoni*, pages 131–143.

Moot, R. and Retoré, C. (2012). *The Logic of Categorial Grammars*. Springer Berlin Heidelberg.

Mu, S.-C. (2008). Typed λ-calculus interpreter in agda. http://www.iis.sinica.edu.tw/~scm/2008/typed-lambda-calculus-interprete/. Accessed: 2014-01-28.

Norell, U. (2009). Dependently typed programming in agda. In *Proceedings of the 4th International Workshop on Types in Language Design and Implementation*, TLDI '09, pages 1–2, New York, NY, USA. ACM.

Parigot, M. (1992). λμ-calculus: An algorithmic interpretation of classical natural deduction. In Voronkov, A., editor, *Logic Programming and Automated Reasoning*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer Berlin Heidelberg.

Restall, G. (2011). Substructural logics. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Summer 2011 edition.

Wadler, P. L. (1993). A Taste of Linear Logic. In *Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science, Gdánsk*, New York, NY. Springer-Verlag.