Mechanised Relation-Algebraic Order Theory in Ordered Categories without Meets

Musa Al-hassy and Wolfram Kahl

McMaster University, Hamilton, Ontario, Canada

29 September 2015 — 15th RAMiCS — Braga, Portugal

This research is supported by the National Science and Engineering Research Council of Canada, NSERC.

Following Up:

RAMiCS 2014:

Future Work

- Prove/implement duality with complete semilattices
- Continue following [Jipsen 2012] to idempotent semirings
- Explore power via meet-free symmetric quotient definition

RAMiCS 2015:

- Staying completely in the meet-free setting
- Assuming OCCs with residuals, symmetric quotients, and direct powers
  (weakening to OSGCs where natural)
- Full development (189 pages (somewhat) literate Agda document) at http://relmics.mcmaster.ca/RATH-Agda/#AContext (GPL v. 3)

Overview

- **Purely OCC-based** characterisation of symmetric quotients
- In OCCs with residuals (without assuming existence of all meets), these symmetric quotients still are meets
- Orders are formalised using symmetric quotients for antisymmetry
- grea, lea, lub, glb are **defined** using symmetric quotients
- Order theory and direct powers "still work"
- Complete lower semilattices with meet-preserving homomorphisms **ACSL**: **abstract**ing *Set* to PowerOCC
- ACSL category is dual to AContext category
  - following Moshier in some places
  - but with all details filled in, **correctly**, **checked by Agda**
  - I would not dare to present this if we had done this only in LaTeX
  - proofs not yet streamlined

Ordered Categories with Converse (OCCs)

OCCs are categories where:

- for $\mathsf{A}\ \mathsf{B} : \mathsf{Obj}$, the "homset" $\mathsf{Hom}\ \mathsf{A}\ \mathsf{B}$ is a poset
- the self-dualising *converse* operator $\_\breve{}$ maps $\mathsf{R} : \mathsf{Mor}\ \mathsf{A}\ \mathsf{B}$ to $\mathsf{R}\breve{} : \mathsf{Mor}\ \mathsf{B}\ \mathsf{A}$
- composition and converse are monotonic

OCCs are a common substructure of allegories and Kleene categories:

- The ordering is primitive, not derived
- Many "typically relation-algebraic" formalisations are already possible
- $\Longrightarrow$ [Kahl-2004, JoRMiCS vol. 1]

Residuals

Left residual / right division:

$$\begin{tikzcd} \mathsf{A} \arrow[r, "\mathsf{S} \,/\, \mathsf{R}"'] \arrow[rr, bend left, "\mathsf{S}"] & \mathsf{B} \arrow[r, "\mathsf{R}"'] & \mathsf{C} \end{tikzcd}$$

$\_/\_ \ : \ \mathsf{Mor\ A\ C} \to \mathsf{Mor\ B\ C} \to \mathsf{Mor\ A\ B}$

$$\mathsf{X} \sqsubseteq \mathsf{S} \,/\, \mathsf{R} \qquad \text{iff} \qquad \mathsf{X} \mathbin{;} \mathsf{R} \sqsubseteq \mathsf{S}$$

Predicate logic: $\forall a : \mathsf{A}, b : \mathsf{B} \bullet a(\mathsf{S} \,/\, \mathsf{R})b \Leftrightarrow (\forall c : \mathsf{B} \bullet b\mathsf{R}c \Rightarrow a\mathsf{S}c)$

Schröder categories / relation algebras: $\mathsf{S} \,/\, \mathsf{R} = \overline{\overline{\mathsf{S}} \mathbin{;} \mathsf{R}^{\smile}}$

Right residual / left division:

$$\begin{tikzcd} \mathsf{A} \arrow[r, "\mathsf{Q}"'] \arrow[rr, bend left, "\mathsf{S}"] & \mathsf{B} \arrow[r, "\mathsf{Q} \,\backslash\, \mathsf{S}"'] & \mathsf{C} \end{tikzcd}$$

$\_\backslash\_ \ : \ \mathsf{Mor\ A\ B} \to \mathsf{Mor\ A\ C} \to \mathsf{Mor\ B\ C}$

$$\mathsf{Y} \sqsubseteq \mathsf{Q} \,\backslash\, \mathsf{S} \qquad \text{iff} \qquad \mathsf{Q} \mathbin{;} \mathsf{Y} \sqsubseteq \mathsf{S}$$

Predicate logic: $\forall b : \mathsf{B}, c : \mathsf{C} \bullet b(\mathsf{Q} \,\backslash\, \mathsf{S})c \Leftrightarrow (\forall a : \mathsf{A} \bullet a\mathsf{Q}b \Rightarrow a\mathsf{S}c)$

Schröder categories / relation algebras: $\mathsf{Q} \,\backslash\, \mathsf{S} = \overline{\mathsf{Q}^{\smile} \mathbin{;} \overline{\mathsf{S}}}$

Symmetric Quotients

Right residual / left division:

$$\begin{tikzcd} A \arrow[r, "Q"'] \arrow[rr, bend left, "S"] & B \arrow[r, "Q \setminus S"'] & C \end{tikzcd}$$

$\_\setminus\_ \; : \; \mathsf{Mor}\ A\ B \to \mathsf{Mor}\ A\ C \to \mathsf{Mor}\ B\ C$

$$Y \sqsubseteq Q \setminus S \qquad \text{iff} \qquad Q \fatsemi Y \sqsubseteq S$$

Predicate logic: $\forall b : B, c : C \bullet b(Q \setminus S)c \Leftrightarrow (\forall a : A \bullet aQb \Rightarrow aSc)$

Schröder categories / relation algebras: $Q \setminus S = \overline{Q^{\smile} \fatsemi \overline{S}}$

Symmetric quotient:

$\_\between\_ \; : \; \mathsf{Mor}\ A\ B \to \mathsf{Mor}\ A\ C \to \mathsf{Mor}\ B\ C$

$$Y \sqsubseteq Q \between S \qquad \text{iff} \qquad Q \fatsemi Y \sqsubseteq S \quad \text{and} \quad Y \fatsemi S^{\smile} \sqsubseteq Q^{\smile}$$

Predicate logic: $\forall b : B, c : C \bullet b(Q \between S)c \Leftrightarrow (\forall a : A \bullet aQb \Leftrightarrow aSc)$

Division allegories: $Q \between S \; = \; Q \setminus S \sqcap Q^{\smile} / S^{\smile}$

Schröder categories / relation algebras: $Q \between S = \overline{Q^{\smile} \fatsemi \overline{S}} \sqcap \overline{\overline{Q^{\smile}} \fatsemi S}$

Residuals in Agda

```
record LeftResOp {i j k₁ k₂ : Level} {Obj : Set i}
                 (base : OrderedSemigroupoid j k₁ k₂ Obj)
                 : Set (i ⊍ j ⊍ k₁ ⊍ k₂) where
  open OrderedSemigroupoid base
  infix 9 _/_
  field
    _/_           : {A B C : Obj}
                  → Mor A C → Mor B C → Mor A B
    /-cancel-outer : {A B C : Obj}
                  → {S : Mor A C} {R : Mor B C}
                  → (S / R) ⨾ R ⊑ S
    /-universal   : {A B C : Obj}
                  → {S : Mor A C} {R : Mor B C} {Q : Mor A B}
                  → Q ⨾ R ⊑ S → Q ⊑ S / R
```

Symmetric Quotients in Agda

```
record SyqOp {i j k₁ k₂ : Level} {Obj : Set i}
             (base : OSGC j k₁ k₂ Obj)
             : Set (i ⊍ j ⊍ k₁ ⊍ k₂) where
  open OSGC base
  infix 9 _╳_                       -- operator precedence level
  field
    _╳_           : {A B C : Obj}
                  → Mor A B → Mor A C → Mor B C
    ╳-cancel-left  : {A B C : Obj}
                  → {Q : Mor A B} {S : Mor A C}
                  → Q ⨾ (Q ╳ S)  ⊑ S
    ╳-cancel-right : {A B C : Obj}
                  → {Q : Mor A B} {S : Mor A C}
                  → (Q ╳ S) ⨾ S ˘ ⊑ Q ˘
    ╳-universal    : {A B C : Obj}
                  → {Q : Mor A B} {S : Mor A C} {R : Mor B C}
                  → Q ⨾ R ⊑ S → R ⨾ S ˘ ⊑ Q ˘ → R ⊑ Q ╳ S
```

Converse of Symmetric Quotients

```
╳-cancel-left  : ... → Q ⨾ (Q ╳ S)  ⊑ S
╳-cancel-right : ... → (Q ╳ S) ⨾ S ˘ ⊑ Q ˘
╳-universal    : ... → Q ⨾ R ⊑ S → R ⨾ S ˘ ⊑ Q ˘ → R ⊑ Q ╳ S

╳-˘-⊑ : {A B C : Obj} {Q : Mor A B} {S : Mor A C}
      → (Q ╳ S) ˘ ⊑ S ╳ Q
╳-˘-⊑ {A} {B} {C} {Q} {S}  =  ╳-universal
  (⊑-begin
       S ⨾ (Q ╳ S) ˘
     ≈˘⟨ ⨾-˘ ⟩
       ((Q ╳ S) ⨾ S ˘) ˘
     ⊑⟨ ˘-monotone  ╳-cancel-right ⟩
       Q ˘ ˘
     ≈⟨ ˘˘ ⟩
       Q
  □)
  (⊑-begin
       (Q ╳ S) ˘ ⨾ Q ˘
     ≈˘⟨ ⨾-˘ ⟩
```

Preorders

$$
\begin{array}{l}
\textbf{record}\ \mathsf{IsPreorder}\ \{\mathsf{A} : \mathsf{Obj}\}\ (\mathsf{E} : \mathsf{Mor\ A\ A}) : \mathsf{Set}\ \mathsf{k}_2\ \textbf{where} \\
\quad \textbf{field}\ \mathsf{refl} \quad : \mathsf{Id} \sqsubseteq \mathsf{E} \qquad\quad \text{-- reflexivity} \\
\qquad\qquad \mathsf{trans} : \mathsf{E} \,;\, \mathsf{E} \sqsubseteq \mathsf{E} \quad \text{-- transitivity}
\end{array}
$$

$$
\begin{array}{l}
\mathsf{ubd}\ \mathsf{lbd} : \{\mathsf{I} : \mathsf{Obj}\} \to \mathsf{Mor\ I\ A} \to \mathsf{Mor\ I\ A} \\
\mathsf{ubd}\ \mathsf{Q} = \mathsf{Q}\,\breve{}\ \backslash\ \mathsf{E} \\
\mathsf{lbd}\ \mathsf{Q} = \mathsf{Q}\,\breve{}\ \backslash\ \mathsf{E}\,\breve{}
\end{array}
$$

$$
\begin{array}{llll}
\multicolumn{4}{l}{\mathsf{gre}\ \mathsf{lea}\ \mathsf{lub}\ \mathsf{glb} : \{\mathsf{I} : \mathsf{Obj}\} \to \mathsf{Mor\ I\ A} \to \mathsf{Mor\ I\ A}} \\
\mathsf{gre}\ \mathsf{Q} = & (\mathsf{E} \,;\, \mathsf{Q}\,\breve{}) & \between\ \mathsf{E} & \\
\mathsf{lea}\ \mathsf{Q} = & (\mathsf{E}\,\breve{} \,;\, \mathsf{Q}\,\breve{}) & \between\ \mathsf{E}\,\breve{} & \\
\mathsf{lub}\ \mathsf{Q} = & \mathsf{ubd}\ \mathsf{Q}\,\breve{} & \between\ \mathsf{E}\,\breve{} & \text{--}\ \approx (\mathsf{E}\,\breve{}\ /\ \mathsf{Q})\ \between\ \mathsf{E}\,\breve{} \\
\mathsf{glb}\ \mathsf{Q} = & \mathsf{lbd}\ \mathsf{Q}\,\breve{} & \between\ \mathsf{E} & \text{--}\ \approx (\mathsf{E}\ /\ \mathsf{Q})\ \between\ \mathsf{E}
\end{array}
$$

Orders

```
record IsOrder {A : Obj} (E : Mor A A) : Set k₂ where
  field refl      : Id ⊑ E                 -- reflexivity
        trans     : E ⨾ E ⊑ E              -- transitivity
```

In division allegories:

```
isAntisymmetric E  =  E ⊓ E ˘ ⊑ Id
preorder-equiv≈⨉ : {A : Obj} {E : Mor A A}
                  → IsPreorder occ E →  E ⊓ E ˘ ≈  E ⨉ E
preorder-equiv≈⨉ {A} {E} E-isPreorder = ≈-begin
      E ⊓ E ˘
    ≈˘⟨ ⊓-cong (preorder-\ refl trans) (preorder-/ ˘-refl ˘-trans) ⟩
      E \ E ⊓ E ˘ / E ˘
    ≈˘⟨ ⨉≈\⊓/ ⟩
      E ⨉ E
    □
    where open IsPreorder occ E-isPreorder
```

```
        antisym : E ⨉ E ⊑ Id             -- antisymmetry
```

Mappings are Fixed-points of lub

Some properties become easier to prove with symmetric quotients:

```
lub-mapping : {I : Obj} {R : Mor I A} → isMapping R → lub R ≈ R
lub-mapping {I} {R} R-map  =  ≈-begin
      lub R
   ≈⟨⟩
      ubd R ˘ ≬ E ˘
   ≈⟨ ≬-cong₁ (˘-cong  (ubd-mapping R-map) ⟨≈≈⟩ ⨾-˘)  ⟩
      (E ˘ ⨾ R ˘) ≬ E ˘
   ≈˘⟨ ≬-in-left R-map ⟩
      R ⨾ (E ˘ ≬ E ˘)
   ≈⟨ ⨾-cong₂  ˘-antisym≈ ⟨≈≈⟩ rightId ⟩
      R
   □
```

lub Q ≈ glb (ubd Q)

```
lub-≈-glb-ubd : {I : Obj} {Q : Mor I A} → lub Q ≈ glb (ubd Q)
lub-≈-glb-ubd {I} {Q} = ≈-begin
    lub Q
  ≈⟨⟩
    ubd Q ˘ ╳ E ˘
  ≈⟨ ⊑-antisym
    (╳-universal
      (⊑-begin
          (E / ubd Q) ⨾ (ubd Q ˘ ╳ E ˘)
        ⊑⟨ ⨾-monotone₂  ˘╳˘-⊑-/ ⟩
          (E / ubd Q) ⨾ (ubd Q / E)
        ⊑⟨ /-cancel-middle ⟨⊑≈⟩ order-/ ⟩
          E
        □)
      (⊑-begin
          (ubd Q ˘ ╳ E ˘) ⨾ E ˘
        ⊑⟨ ⨾-monotone₁  ╳-⊑-\ ⟩
          (ubd Q ˘ \ E ˘) ⨾ E ˘
        ≈⟨ lbd-downclosed ⟩
          ubd Q ˘ \ E ˘
        ≈˘⟨ /-˘ ⟩
          (E / ubd Q) ˘
        □))
    (╳-universal
      (⊑-begin
          ubd Q ˘ ⨾ ((E / ubd Q) ╳ E)
        ⊑⟨ ⨾-monotone₂  (╳-⊑-/ ⟨⊑≈˘⟩ \-˘ ⟨⊑≈⟩ ˘-co
          ubd Q ˘ ⨾ ((E \ E) / ubd Q) ˘
        ⊑⟨ ⨾-˘ ⟨≈˘⊑⟩ ˘-monotone  (/-cancel-outer ⟨⊑
          E ˘
        □)
      ((⊑-begin
          ((E / ubd Q) ╳ E) ⨾ (E ˘) ˘
        ⊑⟨ ⨾-monotone  ╳-⊑-\ (⊑-reflexive ˘˘) ⟩
          ((E / ubd Q) \ E) ⨾ E
        ≈⟨ ⨾-cong₁  \S∘S/∘\S ⟨≈≈⟩ ubd-upclosed ⟩
          ubd Q
        □) ⟨⊑≈˘⟩ ˘˘))
  ⟩
    (E / ubd Q) ╳ E
  ≈˘⟨ ╳-cong₁  lbd-ubd-˘ ⟩
    lbd (ubd Q) ˘ ╳ E
  ≈⟨⟩
    glb (ubd Q)
  □
```

Suborders

```
module SubOrder {A : Obj} {E : Mor A A} (E-isOrder : IsOrder E)
      {Z : Obj} (F : Mapping Z A) (F-inj : isInjective (Mapping.mor F))
  ...
  subOrder-isOrder : IsOrder (F₀ ⨾ E ⨾ F₀ ˘)
  subOrder-isOrder = record {···
    ; antisym = ⊑-begin
          (F₀ ⨾ E ⨾ F₀ ˘) ⊓ (F₀ ⨾ E ⨾ F₀ ˘)
        ≈⟨ ⊓-cong ⨾-assocL ⨾-assocL ⟩
          ((F₀ ⨾ E) ⨾ F₀ ˘) ⊓ ((F₀ ⨾ E) ⨾ F₀ ˘)
        ≈⟨ ⊓-in-left F-isM ⟨≈˘≈˘⟩ ⨾-cong₂ (⊓-M-in-right F-isM) ⟩
          F₀ ⨾ ((F₀ ⨾ E) ⊓ (F₀ ⨾ E)) ⨾ F₀ ˘
        ⊑⟨ retract⊓ rightSupId rightSupId ··· F-unival ··· trans ··· ⟩
          (E ⨾ F₀ ˘) ⊓ (E ⨾ F₀ ˘)
        ≈⟨ ⊓-in-left F-isM ⟨≈˘≈˘⟩ ⨾-cong₂ (⊓-M-in-right F-isM) ⟩
          F₀ ⨾ (E ⊓ E) ⨾ F₀ ˘
        ⊑⟨ ⨾-cong₂ (⨾-cong₁ antisym≈ ⟨≈≈⟩ leftId) ⟨≈⊑⟩ isInjective-to-I F-inj ⟩
          Id
        □
```

$$\begin{tikzcd} & & C_1 \\ A \arrow[r, "H_1\ H_2"] & B \arrow[ur, "F_1\ G_1"] \arrow[dr, "F_2\ G_2"'] & \\ & & C_2 \end{tikzcd}$$

```
retract╳ : {A B C₁ C₂ : Obj}
           {F₁ G₁ : Mor B C₁} {F₂ G₂ : Mor B C₂}
           {H₁ H₂ : Mor A B}
         → F₁ ⊑ G₁
         → F₂ ⊑ G₂
         → H₁ ⨾ G₂ ⨾ F₂ ˘ ⊑ H₂
         → F₁ ⨾ G₁ ˘ ⨾ H₂ ˘ ⊑ H₁ ˘
         → F₁ ⨾ (G₁ ╳ G₂) ⨾ F₂ ˘ ⊑ H₁ ╳ H₂
retract╳ {A} {B} {C₁} {C₂} {F₁} {G₁} {F₂} {G₂} {H₁} {H₂}
         F₁⊑G₁ F₂⊑G₂ H₁⨾G₂⨾F₂˘⊑H₂ F₁⨾G₁˘⨾H₂˘⊑H₁˘ = ╳-universal
           (⊑-begin
              H₁ ⨾ F₁ ⨾ (G₁ ╳ G₂) ⨾ F₂ ˘
           ⊑⟨ ⨾-monotone₂₁ F₁⊑G₁ ⟩
              H₁ ⨾ G₁ ⨾ (G₁ ╳ G₂) ⨾ F₂ ˘
           ⊑⟨ ⨾-monotone₂ (⨾-assocL ⟨≈⊑⟩ ⨾-monotone₁ ╳-cancel-left) ⟩
              H₁ ⨾ G₂ ⨾ F₂ ˘
           ⊑⟨ H₁⨾G₂⨾F₂˘⊑H₂ ⟩
              H₂
           □)
           (⊑-begin
              (F₁ ⨾ (G₁ ╳ G₂) ⨾ F₂ ˘) ⨾ H₂ ˘
           ⊑⟨ ⨾-monotone₁ (⨾-monotone₂₂ (˘-monotone F₂⊑G₂)) ⟩
```

Direct Powers

```
record DirectPower : Set (i ⊍ j ⊍ k₂)  where
  field
    ℙ : Obj → Obj                         -- power object constructor
    ∈ : {A : Obj} → Mor A (ℙ A)           -- membership “relation”
    ∈-extensional : ∈ ╲╱ ∈ ⊑ Id           -- sets defined by extension
    ∈-comprehensive : ∀ {Q} → isTotal (Q ╲╱ ∈)   -- all possible sets

  Ω : {A : Obj} → Mor (ℙ A) (ℙ A)         -- the set inclusion “relation”
  Ω = ∈ \ ∈

  Λ₀ : {I A : Obj} → Mor I A → Mor I (ℙ A)    -- “power transpose”
  Λ₀ R = R ˘ ╲╱ ∈

  Λ : {I A : Obj} → Mor I A → Mapping I (ℙ A)    -- power tr. mapping
  Λ R = record {mor = Λ₀ R; prf = ... }
```

Polarities

Set-theoretically:

$$\begin{array}{rcl} S \uparrow (A) & = & \text{“the } S\text{-}s\text{uccessors of all of } A\text{”} \\ & = & \{s \mid \forall a \in A \,.\, a\, S\, s\} \\ & = & \Lambda(\in \backslash S)(A) \end{array}$$

module _ {A B : Obj} **where**
$\_\!\uparrow$: Mor A B → Mapping ($\mathbb{P}$ A) ($\mathbb{P}$ B)
$\_\!\downarrow$: Mor A B → Mapping ($\mathbb{P}$ B) ($\mathbb{P}$ A)
Galois-↓-↑ : {R : Mor A B} → $\Omega \,⨾\, (\mathsf{R} \downarrow_0)\,\breve{}\; \approx \mathsf{R} \uparrow_0 ⨾\, \Omega\,\breve{}$

$$\begin{array}{llll} \mathsf{S} \uparrow_0 & \approx & \Lambda_0\,(\in \backslash\, \mathsf{S}) & \approx\ (\mathsf{S}\,\breve{}\,/ \in\breve{}\,) \between \in \\ \mathsf{S} \downarrow_0 & \approx & \Lambda_0\,(\in \backslash\, \mathsf{S}\,\breve{}\,) & \approx\ (\mathsf{S} / \in\breve{}\,) \between \in \\ \mathsf{S} \downarrow\uparrow_0 & \approx & \mathsf{S} \downarrow_0 ⨾\, \mathsf{S} \uparrow_0 & \approx\ (\mathsf{S}\,\breve{}\,/ (\in \backslash\, \mathsf{S}\,\breve{}\,)) \between \in \\ \mathsf{S} \uparrow\downarrow_0 & \approx & \mathsf{S} \uparrow_0 ⨾\, \mathsf{S} \downarrow_0 & \approx\ (\mathsf{S} / (\in \backslash\, \mathsf{S})) \between \in \\ & & \mathsf{S} \uparrow_0 ⨾\, (\mathsf{S} \downarrow_0)\,\breve{} & \approx\ (\mathsf{S}\,\breve{}\,/ \in\breve{}\,) \between (\mathsf{S} / \in\breve{}\,) \\ & & \mathsf{lbd}\,(\mathsf{R} ⨾\, \mathsf{S} \downarrow\uparrow_0)\,\breve{} & \approx\ ((\in \backslash\, \mathsf{S}\,\breve{}\,) / (\in \backslash\, \mathsf{S}\,\breve{}\,)) / \mathsf{R} \end{array}$$

Complete Semilattices

```
record ACSL : Set (i ⊍ j ⊍ k₂) where
  field Carrier   : Obj
        ≼         : Mor Carrier Carrier
        ≼-isOrder : IsOrder ≼
  open IsOrder ≼-isOrder                  -- to make in particular glb available
  field glb-total : {I : Obj} (R : Mor I Carrier) → isTotal (glb R)

record ACSLHom (A B : ACSL) : Set (i ⊍ j ⊍ k₁ ⊍ k₂) where
  field map : Mapping A.Carrier B.Carrier
  map₀ : Mor A.Carrier B.Carrier
  map₀ = Mapping.mor map
  field monotone   : A.≼ ⨾ map₀     ⊑ map₀ ⨾ B.≼
        continuous : {I : Obj} {S : Mor I A.Carrier}
                   → A.glb S ⨾ map₀ ≈ B.glb (S ⨾ map₀)
```

Preparing Monotony from Continuity

The glb of the image of the “up-set” of x is the image of x:

```
glb-≼⨾continuous : B.glb (A.≼ ⨾ map₀) ≈ map₀ ˆ!
glb-≼⨾continuous = ≈-begin
    B.glb (A.≼ ⨾ map₀)
  ≈˘⟨ continuous ⟩
    A.glb A.≼ ⨾ map₀ ˆ!
  ≈⟨ ⨾-cong₁ A.glb-order ⟨≈≈⟩ leftId ⟩
    map₀
  □
```

Monotony from Continuity

```
monotone′ = ⊑-begin
     A.≼ ⨾ map₀
   ⊑⟨ proj₁ (mappingTotal map) ⟨⊑≈⟩ ⨾-assoc ⟩
     map₀ ⨾ map₀ ˘ ⨾ A.≼ ⨾ map₀
   ≈⟨ ⨾-cong₂₁ (˘-cong glb-≼⨾continuous ⟨≈˘≈⟩ (╳-˘ ⟨≈≈⟩ ╳-cong₂ ˘\˘-˘))
     map₀ ⨾ (B.≼ ╳ (B.≼ / (A.≼ ⨾ map₀))) ⨾ A.≼ ⨾ map₀
   ⊑⟨ ⨾-monotone₂ (\-universal (⊑-begin
        B.≼ ⨾ (B.≼ ╳ (B.≼ / (A.≼ ⨾ map₀))) ⨾ A.≼ ⨾ map₀
      ⊑⟨ ⨾-assocL ⟨≈⊑⟩ ⨾-monotone₁ ╳-cancel-left ⟩
        (B.≼ / (A.≼ ⨾ map₀)) ⨾ A.≼ ⨾ map₀
      ⊑⟨ /-cancel-outer ⟩
        B.≼
      □)) ⟩
     map₀ ⨾ (B.≼ \ B.≼)
   ≈⟨ ⨾-cong₂ B.order-\ ⟩
     map₀ ⨾ B.≼
   □
```

Contexts — see RAMiCS 2014

An “abstract” *context* is merely a typed “relation”:

```
record AContext : Set (i ⊍ j) where
  field ent : Obj                          -- “entities”
        att : Obj                          -- “attributes”
        inc : Mor ent att                  -- “incidence”

record AContextHom (A B : AContext) : Set (i ⊍ j ⊍ k₁ ⊍ k₂) where
  field mor : Mor A.ent B.att
        srcCompat : mor ↓ ⨾₁ A.inc ↑ ⨾₁ A.inc ↓ ≈₁ mor ↓
        trgCompat : B.inc ↓ ⨾₁ B.inc ↑ ⨾₁ mor ↓ ≈₁ mor ↓
```

$$
\begin{tikzcd}
\mathbb{P}\,\mathsf{A.att} \arrow[r, "\mathsf{A.inc}\downarrow"] & \mathbb{P}\,\mathsf{A.ent} & \mathbb{P}\,\mathsf{B.att} \arrow[l, dotted, "\mathsf{mor}\downarrow"'] \\
\mathbb{P}\,\mathsf{A.ent} \arrow[u, "\mathsf{A.inc}\uparrow"] & \mathbb{P}\,\mathsf{B.att} \arrow[l, dotted, "\mathsf{mor}\downarrow"'] \arrow[r, "\mathsf{B.inc}\downarrow"] \arrow[u, dotted, "\mathsf{mor}\downarrow"] & \mathbb{P}\,\mathsf{B.ent} \arrow[u, "\mathsf{B.inc}\uparrow"]
\end{tikzcd}
$$

Duality between Contexts and CSLs

- For $\mathsf{A} : \mathsf{AContext}$, construct an ACSL:
  - Carrier made up of $\mathsf{A.inc}{\uparrow\downarrow}$-closed subsets of $\mathsf{A.ent}$
  - $\mathsf{R} : \mathsf{AContextHom\ A\ B}$ mapped to $\mathsf{B.inc}\uparrow ⨾ \mathsf{R.mor}\downarrow$ — contravariant lattice hom., from $\mathbb{P}\ \mathsf{B.ent}$ to $\mathbb{P}\ \mathsf{A.ent}$,
  - In detail, using $\uparrow\downarrow$-image injections Γ:

$$\Phi_0 : \mathsf{Mor\ B.}{\uparrow\downarrow}\text{-}\mathsf{image\ A.}{\uparrow\downarrow}\text{-}\mathsf{image}$$
$$\Phi_0 = \mathsf{B.}\Gamma ⨾ \mathsf{B.inc}\uparrow_0 ⨾ \mathsf{R.mor}\downarrow_0 ⨾ \mathsf{A.}\Gamma\,\breve{}$$

- From $\mathsf{A} : \mathsf{ACSL}$ to "standard polarity":

$$\mathsf{fromACSL} : \mathsf{ACSL} \to \mathsf{AContext}$$
$$\mathsf{fromACSL\ A} = \textbf{record}\ \{\mathsf{ent} = \mathsf{A.Carrier}; \mathsf{att} = \mathsf{A.Carrier}; \mathsf{inc} = \mathsf{A.}{\preccurlyeq}\}$$

  - CSL morphism $\Phi : \mathsf{A} \to \mathsf{B}$ mapped to $\mathsf{B.}{\preccurlyeq} ⨾ \Phi_0\,\breve{}$

Arbitrary Intersections of Closed Sets are Closed

```
glb-closed-⊑ : {I : Obj} {Q : Mor I A} → Q ⨾ C ≈ Q → glb Q ⨾ C ⊑ glb Q
glb-closed-⊑ {I} {Q} Q⨾C≈Q = ˘≬-universal (⊑-begin
        lbd Q ˘ ⨾ (lbd Q ˘ ≬ E) ⨾ C
     ⊑⟨ ⨾-assocL ⟨≈⊑⟩ ⨾-monotone₁ ≬-cancel-left ⟩
        E ⨾ C
     ⊑⟨ ⨾-monotone₂ C⊑E ⟨⊑⊑⟩ trans ⟩
        E
   □) (⨾-assoc ⟨≈⊑⟩ \-universal ((⊑-begin
        Q ˘ ⨾ (lbd Q ˘ ≬ E) ⨾ (C ⨾ E ˘)
     ⊑⟨ ⨾-cong₁ (˘-cong Q⨾C≈Q ⟨≈˘≈⟩ ⨾-˘) ⟨≈⊑⟩ ⨾-monotone₂₁ ˘≬-⊑-/ ⟩
        (C ˘ ⨾ Q ˘) ⨾ ((Q ˘ \ E ˘) / E ˘) ⨾ (C ⨾ E ˘)
     ⊑⟨ ⨾-₂₂assoc₁₂₁ ⟨≈⊑⟩ ⨾-monotone₂₁ /-outer-⨾ ⟩
        C ˘ ⨾ ((Q ˘ ⨾ (Q ˘ \ E ˘)) / E ˘) ⨾ (C ⨾ E ˘)
     ⊑⟨ ⨾-monotone₂₁ (/-monotone \-cancel-outer ⟨⊑≈⟩ order˘-/) ⟩
        C ˘ ⨾ E ˘ ⨾ (C ⨾ E ˘)
     ⊑⟨ ⨾-monotone₁&₂₁ C-monotone˘ ⟩
        E ˘ ⨾ C ˘ ⨾ (C ⨾ E ˘)
     ⊑⟨ ⨾-monotone₂ (⨾-assocL ⟨≈⊑⟩ proj₁ C.unival) ⟩
        E ˘ ⨾ E ˘
     ⊑⟨ ˘-trans ⟩
        E ˘
   □)))
```

Source Compatibility of fromACSLHom

```
fromACSLHom : ACSLHom A B → AContextHom (fromACSL B) (fromACSL A)
fromACSLHom Φ = record
  {mor = ≼B ⨾ Φ₀ ˘                  -- Φ₀ is the underlying mapping of Φ
  ;srcCompat = ≈-begin
        (≼B ⨾ Φ₀ ˘) ↓₀⨾ ≼B ↑↓₀
     ≈⟨ ⨾-cong₂ ↑↓≈╳ ⟩
        (≼B ⨾ Φ₀ ˘) ↓₀ ⨾ ((≼B / (∈ \ ≼B)) ╳ ∈)
     ≈⟨ ╳-in-left (Mapping.prf ((≼B ⨾ Φ₀ ˘) ↓)) ⟩
        ((≼B / (∈ \ ≼B)) ⨾ (≼B ⨾ Φ₀ ˘) ↓₀ ˘) ╳ ∈
     ≈⟨ ╳-cong₁ (/-inner-⨾ (Mapping.prf ((≼B ⨾ Φ₀ ˘) ↓))) ⟩
        (≼B / ((≼B ⨾ Φ₀ ˘) ↓₀ ⨾ (∈ \ ≼B))) ╳ ∈
     ≈⟨ ╳-cong₁ (/-cong₂ ↓⨾∈\) ⟩
        (≼B / (((≼B ⨾ Φ₀ ˘) / ∈ ˘) \ ≼B)) ╳ ∈
     ≈˘⟨ ╳-cong₁ (/-cong₂ (\-cong₁ (/-flip (Mapping.prf Φ)))) ⟩
        (≼B / ((≼B / (∈ ˘ ⨾ Φ₀)) \ ≼B)) ╳ ∈
     ≈⟨ ╳-cong₁ S/∘\S∘S/ ⟩
        (≼B / (∈ ˘ ⨾ Φ₀)) ╳ ∈
     ≈⟨ ╳-cong₁ (/-flip (Mapping.prf Φ)) ⟩
        ((≼B ⨾ Φ₀ ˘) / ∈ ˘) ╳ ∈
     ≈˘⟨ ↓≈╳ ⟩
        (≼B ⨾ Φ₀ ˘) ↓₀
     □
  ;trgCompat = ...
```

Naturality for Ctx→CSL→Ctx

```
C-naturality : {A B : AContext} → {F : AContextHom A B}
             → (fromACSLHom (toACSLHom F) ⨾₂ C {B}) ≈₂ (C {A} ⨾₂ F)
C-naturality {A} {B} {F} = ˘-cong (≈-begin
    Λ₀ Id ⨾ (B.R ↓₀ ⨾ B.≼ ↑₀ ⨾ (A.≼ ⨾ F₁.map₀ ˘) ↓₀) ⨾ ∈ ˘
  ≈⟨ ⨾-cong₂ (⨾-assoc₃₊₁ ⟨≈≈⟩ ⨾-cong₂₂ ↓⨾∈˘) ⟩
    Λ₀ Id ⨾ B.R ↓₀ ⨾ B.≼ ↑₀ ⨾ (∈ \ (A.≼ ⨾ F₁.map₀ ˘) ˘)
  ≈⟨ ⨾-assocL ⟨≈≈⟩ ⨾-cong ΛId-⨾-↓ ↑⨾∈\ ⟩
    Λ₀ (B.R ˘) ⨾ ((∈ \ B.≼) ˘ \ (A.≼ ⨾ F₁.map₀ ˘) ˘)
  ≈⟨ \-inner-⨾ Λ-isMapping ⟨≈≈⟩ \-cong₁ (⨾-˘ ⟨≈˘≈⟩ ˘-cong Λ-˘-⨾-∈\) ⟩
    (B.R \ B.≼) ˘ \ (A.≼ ⨾ F₁.map₀ ˘) ˘
  ≈⟨ /-˘ ⟨≈˘≈⟩ ˘-cong (≈-begin
        (A.≼ ⨾ F₁.map₀ ˘) / (B.R \ B.≼)
      ≈˘⟨ /-flip (Mapping.prf F₁.map) ⟩
        A.≼ / ((B.R \ B.≼) ⨾ F₁.map₀)
      ≈⟨ /-cong₂ (⨾-cong₁ (B.Q˘⨾-\-Q˘⨾ ⟨≈≈⟩ \-cong (⨾-assocL ⟨≈≈⟩ ⨾-cong₁ ↑↓⨾↑ ⟨≈≈⟩ ↑⨾∈˘) (B.inc↑↓-⨾-Ω-⨾-Q ⟨≈≈⟩ B.Ω⨾
        A.≼ / (((∈ \ B.inc) \ (∈ \ (∈ ⨾ B.Q))) ⨾ F₁.map₀)
      ≈⟨ /-cong₂ (⨾-cong₁ (∈\-\-∈\ ⟨≈≈˘⟩ \-outer-⨾-≈-˘M B.Q˘-isMapping) ⟨≈≈⟩ ⨾-assoc) ⟩
        A.≼ / ((B.inc \ ∈) ⨾ B.Q ⨾ F₁.map₀)
      ≈⟨⟩
        A.≼ / ((B.inc \ ∈) ⨾ B.Q ⨾ B.Q˘ ⨾ B.inc ↑₀ ⨾ F.mor ↓₀ ⨾ A.Q)
      ≈⟨ /-cong₂ (⨾-cong₂ (⨾-cong₁&₂₁ (B.factors ⟨≈≈˘⟩ ran↓-≈-ran↑↓)) ⟨≈≈⟩ ⨾-assocL) ⟩
        A.≼ / (((B.inc \ ∈) ⨾ B.inc ↓₀ ˘) ⨾ B.inc ↓₀ ⨾ B.inc ↑₀ ⨾ F.mor ↓₀ ⨾ A.Q)
      ≈⟨ /-cong₂ (⨾-cong (\-outer-⨾-≈ (Mapping.prf (B.inc ↓)) ⟨≈≈⟩ \-cong₂ (⨾˘-˘ ⟨≈˘≈⟩ ˘-cong ↓⨾∈˘ ⟨≈≈⟩ \˘-˘))
                             (⨾-assocL ⟨≈≈⟩ ⨾-assocL ⟨≈≈⟩ ⨾-cong₁ F.trgCompat)) ⟩
        A.≼ / ((B.inc \ (B.inc / ∈ ˘)) ⨾ F.mor ↓₀ ⨾ A.Q)
      ≈⟨ /-cong (⨾-assocL ⟨≈≈⟩ ⨾-cong₁ A.Q˘⨾Ω) ⨾-assocL ⟩
        ((A.S \ ∈) ⨾ A.Q) / (((B.inc \ (B.inc / ∈ ˘)) ⨾ F.mor ↓₀) ⨾ A.Q)
      ≈⟨ /-flip-˘ A.Q˘-isMapping ⟨≈˘≈⟩ /-cong₂ (⨾-assoc ⟨≈≈⟩ ⨾-cong₂ A.factors ⟨≈≈⟩ ⨾-assoc ⟨≈≈⟩ ⨾-cong₂ F.srcCompat′
        (A.S \ ∈) / ((B.inc \ (B.inc / ∈ ˘)) ⨾ F.mor ↓₀)
      ≈⟨ /-flip (Mapping.prf (F.mor ↓)) ⟨≈≈⟩ /-cong₁ (\-outer-⨾-≈ (Mapping.prf (F.mor ↓))) ⟩
        (A.S \ (∈ ⨾ F.mor ↓₀ ˘)) / (B.inc \ (B.inc / ∈ ˘))
```

Conclusion

- Abstract formalisation of Context and CSL categories and proof of duality **only need OCC with residuals, syq, direct powers**
- Agda used as **"mechanised mathematical notation"**:

  natural formalisations, confident proofs
- The **abstract algebraic style** plays to the strengths of Agda
- Constructing categories over DivOCCs may require copatterns and "no-eta" for ease of typechecking

Future Work

- Streamline proofs
- Replace "standard polarity"
- Continue following [Jipsen 2012] to idempotent semirings
- Explore automation of meet-free residual reasoning

Path-Compatible Residuals

Question: Is anybody aware of notational experiments with **"composition-like residuals"**?

module _ {A B C : Obj} {Q : Mor A B} {R : Mor B C} **where**

Leftwards implication:

$\_\Leftarrow\!;\_ \;:\; \mathsf{Mor}\ A\ B \to \mathsf{Mor}\ B\ C \to \mathsf{Mor}\ A\ C$

$X \sqsubseteq Q \Leftarrow\!; R$ iff $X ; R\breve{\ } \sqsubseteq Q$

Schröder categories / relation algebras: $Q \Leftarrow\!; R = \overline{\overline{Q} ; R}$

Rightwards implication:

$\_;\!\Rightarrow\_ \;:\; \mathsf{Mor}\ A\ B \to \mathsf{Mor}\ B\ C \to \mathsf{Mor}\ A\ C$

$X \sqsubseteq Q ;\!\Rightarrow R$ iff $Q\breve{\ } ; X \sqsubseteq R$

Schröder categories / relation algebras: $Q ;\!\Rightarrow R = \overline{Q ; \overline{R}}$

"bi-implicational composition":

$\_\Leftarrow\!;\!\Rightarrow\_ \;:\; \mathsf{Mor}\ A\ B \to \mathsf{Mor}\ B\ C \to \mathsf{Mor}\ A\ C$

$X \sqsubseteq Q \Leftarrow\!;\!\Rightarrow R$ iff $Q\breve{\ } ; X \sqsubseteq R$ and $X ; R\breve{\ } \sqsubseteq Q$

Schröder categories / RA: $Q \Leftarrow\!;\!\Rightarrow R = \overline{Q ; \overline{R}} \sqcap \overline{\overline{Q} ; R}$