Types of Algebraic Structures in Proof Assistant Systems

Akshobhya Katte Madhusudana
Under the supervision of Dr. Jacques Carette

McMaster University

November 12^{th}, 2023

Table of Contents

Table of Contents

Research outline

1. What is the current coverage of algebraic structures in proof assistant systems?
2. How to characterize types of algebraic structures in Agda?
3. Define constructs of algebraic structures with proofs to their properties in Agda
4. Abstract out the problems faced during the characterization of algebraic structures and analyze each problem to provide plausible solutions.

Table of Contents

Algebraic structures in proof systems

1. A survey of coverage of algebraic structures in proof systems will help to identify the gaps in the system.
2. Survey on standard libraries of four proof assistant systems Agda, Lean, Idris, and Coq.
3. Create a web crawler to capture definitions of algebraic structures.
4. Create a clickable table that takes to the definition in the source library.

Table of Contents

Theory of Quasigroup and Loop

Quasigroup is a set equipped with binary operations that satisfy the following equation.

$$y = x{\cdot}(x \backslash y)$$

$$y = x \backslash (x{\cdot}y)$$

$$y = (y/x){\cdot}x$$

$$y = (y{\cdot}x)/x$$

Loop is a quasigroup with identity:

$$x{\cdot}e = e{\cdot}x = x$$

```
record IsQuasigroup (· \\ // : Op₂ A) : Set
↪ (a ⊔ ℓ) where
  field
    isMagma        : IsMagma ·
    \\-cong        : Congruent₂ \\
    //-cong        : Congruent₂ //
    leftDivides    : LeftDivides · \\
    rightDivides   : RightDivides · //
  open IsMagma isMagma public
```

Types of Quasigroup

A loop is called a *right bol loop* if it satisfies the identity

$$((z \cdot x) \cdot y) \cdot x \;=\; z \cdot ((\, x \cdot y) \cdot x)$$

A loop is called a *left bol loop* if it satisfies the identity

$$x \cdot (y \cdot (x \cdot z)) \;=\; (x \cdot (y \cdot x)) \cdot z$$

A loop is called *middle bol loop* if it satisfies the identity

$$(z \cdot x) \cdot (y \cdot z) \;=\; z \cdot ((x \cdot y) \cdot z)$$

A left-right bol loop is called a *moufang loop* if it satisfies identity

$$(z \cdot x) \cdot (y \cdot z) \;=\; z \cdot ((x \cdot y) \cdot z)$$

Quasigroup homomorphism

A Quasigroup homomorphism $f : (Q_1, \cdot, \backslash\backslash, //) \to (Q_2, \circ, \backslash, /)$

- f preserves the binary operation: $f(x \cdot y) = f(x) \circ f(y)$
- f preserves the left division operation : $f(x \backslash\backslash y) = f(x) \backslash f(y)$
- f preserves the right division operation: $f(x//y) = f(x)/f(y)$

```
record IsQuasigroupHomomorphism (⟦_⟧ : A → B) : Set (a ⊔ ℓ₁ ⊔ ℓ₂) where
 field
   isRelHomomorphism : IsRelHomomorphism _≈₁_ _≈₂_ ⟦_⟧
   ∙-homo            : Homomorphic₂ ⟦_⟧ _∙₁_ _∙₂_
   \\-homo           : Homomorphic₂ ⟦_⟧ _\\₁_ _\\₂_
   //-homo           : Homomorphic₂ ⟦_⟧ _//₁_ _//₂_

 open IsRelHomomorphism isRelHomomorphism public
  renaming (cong to ⟦⟧-cong)
```

Properties of Quasigroup

Properties of cancellative quasigroups are used in cryptographic protocols. Properties such as left and right cancellation can be used to ensure the confidentiality of data during encryption and decryption

```
cancelˡ : LeftCancellative _·_
cancelˡ x y z eq = begin
  y            ≈⟨ sym( leftDividesʳ x y) ⟩
  x \\ (x · y) ≈⟨ \\-congˡ eq ⟩
  x \\ (x · z) ≈⟨ leftDividesʳ x z ⟩
  z            ∎
```

Properties of middle bol loop

1. $x \cdot ((y \cdot x) \backslash x) = y \backslash x$
2. $x \cdot ((x \cdot z) \backslash x) = x / z$
3. $x \cdot (z \backslash x) = (x / z) \cdot x$
4. $(x / (y \cdot z)) \cdot x = (x / z) \cdot (y \backslash x)$
5. $(x / (y \cdot x)) \cdot x = y \backslash x$
6. $(x / (x \cdot z)) \cdot x = x / z$

Properties of Moufang loop:

1. Moufang loop is alternative.
2. Moufang loop is flexible.
3. $z \cdot (x \cdot (z \cdot y)) = ((z \cdot x) \cdot z) \cdot y$
4. $x \cdot (z \cdot (y \cdot z)) = ((x \cdot z) \cdot y) \cdot z$
5. $z \cdot ((x \cdot y) \cdot z) = (z \cdot (x \cdot y)) \cdot z$

Table of Contents

Semigroup

A Semigroup is a magma with associativity:

$$x \cdot (y \cdot z) \; = \; (x \cdot y) \cdot z$$

```
record IsSemigroup (· : Op₂ A) : Set (a ⊔ ℓ) where
 field
   isMagma : IsMagma ·
   assoc   : Associative ·

 open IsMagma isMagma public
```

A commutative semigroup is a semigroup with commutativity:

$$x \cdot y \; = \; y \cdot x$$

Ring

Ring $(R, +, *, ^{-1}, 0, 1)$

- $(R, +, ^{-1}, 0)$ is an Abelian Group:
  - Associativity: $\forall x, y, z \in R, x + (y + z) = (x + y) + z$
  - commutativity : $\forall x, y \in R, (x + y) = (y + x)$
  - Identity: $\forall x \in R, (x + 0) = x = (0 + x)$
  - Inverse: $\forall x \in R, (x + x^{-1}) = 0 = (x^{-1} + x)$
- $(R, *, 1)$ is a monoid
  - Associativity: $\forall x, y, z \in R, x * (y * z) = (x * y) * z$
  - Identity: $\forall x, y \in R, (x * 1) = x = (1 * x)$
- Multiplication distributes over addition:
  $\forall x, y, z \in R, (x * (y + z)) = (x * y) + (x * z)$ and
  $(x + y) * z = (x * z) + (y * z)$
- Annihilating zero: $\forall x \in R, (x * 0) = 0 = (0 * x)$

Properties of types of Ring

The properties of rings are used in studying number theory and algebraic geometry, where they are used to study algebraic curves, surfaces, and other geometric objects

$$\text{if } x + x = 0 \text{then } x = 0$$

```
x+x≈x⇒x≈0 : ∀ x → x + x ≈ x → x ≈ 0#
x+x≈x⇒x≈0 x eq = begin
  x           ≈⟨ sym(+-identityʳ x) ⟩
  x + 0#      ≈⟨ +-congˡ (sym (-‿inverseʳ x)) ⟩
  x + (x - x) ≈⟨ sym (+-assoc x x (- x)) ⟩
  x + x - x   ≈⟨ +-congʳ(eq) ⟩
  x - x       ≈⟨ -‿inverseʳ x ⟩
  0#          ∎
```

Table of Contents

Idempotent semiring

An idempotent semiring $(S, +, *, 0, 1)$:

- $(S, +, 0)$ is a commutative monoid:
  - Associativity: $\forall x, y, z \in S, x + (y + z) = (x + y) + z$
  - Identity: $\forall x \in S, (x + 0) = x = (0 + x)$
  - Commutativity: $\forall x, y \in S, (x + y) = (y + x)$
- $(S, *, 1)$ is a monoid:
  - Associativity: $\forall x, y, z \in S, x * (y * z) = (x * y) * z$
  - Identity: $\forall x \in S, (x * 1) = x = (1 * x)$
- Idempotent: $\forall x \in S, (x + x) = x$
- Multiplication distributes over addition: $\forall x, y, z \in S, (x * (y + z)) = (x * y) + (x * z)$ and $(x + y) * z = (x * z) + (y * z)$
- Annihilating zero: $\forall x \in S, (x * 0) = 0 = (0 * x)$

Kleene Algebra

A Kleene algebra is an idempotent semiring with unary * operator that satisfies:

$$1 + (x \cdot (x^*)) \leq x^*$$

$$1 + (x^*) \cdot x \leq x^*$$

$$\text{If } b + a \cdot x \leq x \text{ then, } (a^*) \cdot b \leq x$$

$$\text{If } b + x \cdot a \leq x \text{ then, } b \cdot (a^*) \leq x$$

```
record IsKleeneAlgebra (+ * : Op₂ A) (⋆ : Op₁ A) (0# 1# : A) : Set (a ⊔
↪ ℓ) where
  field
    isIdempotentSemiring  : IsIdempotentSemiring + * 0# 1#
    starExpansive         : StarExpansive 1# + * ⋆
    starDestructive       : StarDestructive + * ⋆

  open IsIdempotentSemiring isIdempotentSemiring public
```

Properties of Kleene Algebra

Applications of properties of Kleene Algebra are found in the development of pattern-matching algorithms in text processing and computational linguistics and regular expressions.

```
1+x⋆≈x⋆ : ∀ x → 1# + x ⋆ ≈ x ⋆
1+x⋆≈x⋆ x = begin
  1# + x ⋆                          ≈⟨ +-congˡ (sym(starExpansiveʳ x)) ⟩
  1# + (1# + x ∗ x ⋆ + x ⋆)         ≈⟨ +-congˡ (+-assoc 1# (x ∗ x ⋆) (x ⋆)) ⟩
  1# + (1# + (x ∗ x ⋆ + x ⋆))       ≈⟨ sym(+-assoc 1# 1# (x ∗ x ⋆ + x ⋆)) ⟩
  1# + 1# + (x ∗ x ⋆ + x ⋆)         ≈⟨ +-congʳ (+-idem 1#) ⟩
  1# + (x ∗ x ⋆ + x ⋆)              ≈⟨ sym(+-assoc 1# (x ∗ x ⋆) (x ⋆) ) ⟩
  1# + x ∗ x ⋆ + x ⋆                ≈⟨ starExpansiveʳ x ⟩
  x ⋆                               ∎
```

Table of Contents

Ambiguity and Equivalent

1. Ambiguity in naming e.g. Ring and Rng, Nearring (*-semigroup/*-monoid).
2. Equivalent but structurally different e.g. Quasigroups

A quasigroup with Latin square property is a type (2) algebra.

$$a \cdot x = b$$

$$y \cdot a = b$$

A quasigroup with division operation is a type (2,2,2) algebra

$$y = x \cdot (x \backslash y)$$

$$y = x \backslash (x \cdot y)$$

$$y = (y / x) \cdot x$$

$$y = (y \cdot x) / x$$

Redundant field

Duplicate field: e.g. semiring (+-commutativeMonoid and *-monoid)

```
record IsSemiringWithoutOne (+ * : Op₂ A) (0# : A) : Set (a ⊔ ℓ) where
field
  +-isCommutativeMonoid : IsCommutativeMonoid + 0#
  *-cong                : Congruent₂ *
  *-assoc               : Associative *
  distrib               : * DistributesOver +
  zero                  : Zero 0# *

open IsCommutativeMonoid +-isCommutativeMonoid public
```

Equivalent and Identical

1. Equivalent structures e.g. Bounded semilattice and Idempotent commutative monoid
2. Identical structures e.g. Nearring (+-group, *-monoid)

```
record IsNearring (+ * : Op₂ A) (0# 1# : A) (_⁻¹ : Op₁ A) : Set (a ⊔ ℓ)
↪ where
  field
    isQuasiring : IsQuasiring + * 0# 1#
    +-inverse   : Inverse 0# _⁻¹ +
    ⁻¹-cong     : Congruent₁ _⁻¹

  open IsQuasiring isQuasiring public

  +-isGroup : IsGroup + 0# _⁻¹
  +-isGroup = record
    { isMonoid = +-isMonoid
    ; inverse = +-inverse
    ; ⁻¹-cong = ⁻¹-cong
    }
```

Table of Contents

Conclusion

To sum up, we...

1. Set the scope by doing a survey
2. Study select subset of types of algebraic structures in Agda
3. Analyze five problems that we encountered and provide plausible solution.