Formalizing Lindenbaum-Tarski algebra for propositional logic in Cubical Agda

Caroline Roos

Supervisor: Anders Mörtberg

Degree project

Department of Mathematics
Stockholm University

June 13, 2023

Outline of talk

1. Introduction and motivation
2. Agda proof assistant
3. Classical propositional logic and provability in Agda
4. Constructing the Lindenbaum-Tarski algebra using set quotioents from Cubical Agda
5. Proving the Lindenbaum-Tarski algebra is a Boolean algebra
6. Soundness

Table of Contents

1 Introduction and motivation

2 Agda proof assistant

3 Classical propositional logic and provability in Agda

4 Constructing the Lindenbaum-Tarski algebra using set quotioents from Cubical Agda

5 Proving the Lindenbaum-Tarski algebra is a Boolean algebra

6 Soundness

Lindenbaum-Tarski algebra of propositional logic

- The Lindenbaum-Tarski algebra is the quotient algebra we obtain when quotienting the algebra of formulas by an equivalence relation defined in terms of provability.
- The algebraic operations in the Lindenbaum-Tarski algebra are derived from the logical connectives present in the underlying logical system.
- These operations allow for the manipulation of formulas within the algebraic structure.

Formalizing mathematics

Why formalize?

Benefits of formalizing mathematics:

- Ensure correctness
- Detect errors in tradional mathematical proofs

There are many proof assistants out there
Coq, **Agda**, Lean, Idris, ...

Table of Contents

1 Introduction and motivation

2 Agda proof assistant

3 Classical propositional logic and provability in Agda

4 Constructing the Lindenbaum-Tarski algebra using set quotioents from Cubical Agda

5 Proving the Lindenbaum-Tarski algebra is a Boolean algebra

6 Soundness

Propositions as types

- Agda allows us to encode mathematical propositions as types and their proofs as programs.
- The propositions-as-types interpretation is the direct relationship between computer programs and mathematical proofs.

Prop	Type
$\top$	unit
$\bot$	void
$\phi_1 \wedge \phi_2$	$\tau_1 \times \tau_2$
$\phi_1 \supset \phi_2$	$\tau_1 \to \tau_2$
$\phi_1 \vee \phi_2$	$\tau_1 + \tau_2$

Agda proof assistant

Defining a datatype in Agda

```
data Bool : Type where
  true : Bool
  false : Bool
```

Bool is the name of the datatype, and **true** and **false** are its constructors.

Agda proof assistant

Functions over datatypes can be defined using pattern matching.

```
not : Bool → Bool
not true = false
not false = true
```

The type of **not** is defined as a function from **Bool** to **Bool**. The function is then defined by pattern matching on the arguments.

Agda proof assistant

A **dependent type** is a type that depends on elements of another type.
For example, the polymorphic identity function:

$$\text{id} : (A : \text{Type}) \to A \to A$$
$$\text{id}\ A\ x = x$$

In Agda it is possible to use implicit arguments.

$$\text{id'} : \{A : \text{Type}\} \to A \to A$$
$$\text{id'}\ x = x$$

Agda will try to infer the type for us.

Cubical Agda

#TODO: Redo this one
Cubical Agda is an extension of Agda that incorporates features of cubical type theory.

It has native support for set quotients!

Note: We are using the agda/cubical library, which is the standard library for Cubical Agda.

Table of Contents

1 Introduction and motivation

2 Agda proof assistant

3 Classical propositional logic and provability in Agda

4 Constructing the Lindenbaum-Tarski algebra using set quotioents from Cubical Agda

5 Proving the Lindenbaum-Tarski algebra is a Boolean algebra

6 Soundness

Formalizing propositional logic

The set of well formed formulas is defined inductively:

- any propositional constant $p_0, p_1, \ldots, p_n$ is a well formed formula,
- $\top$ and $\bot$ are well formed formulas,
- if p_i and p_j are well formed formulas, then so are

$$p_i \wedge p_j \quad p_i \vee p_j \quad \neg p_i$$

```
data Formula : Type where
  const : ℕ → Formula
  _∧_ : Formula → Formula → Formula
  _∨_ : Formula → Formula → Formula
  ¬_  : Formula → Formula
  ⊥   : Formula
  ⊤   : Formula
```

Formalizing propositional logic

A context is a set of sentences defined inductively as:

- The empty set is a context
- If Γ a context, then $\Gamma \cup \phi$ a context

For all contexts Γ and all formulas ϕ, ψ:

- $\phi \in \Gamma \cup \{\phi\}$
- If $\phi \in \Gamma$, then $\phi \in \Gamma \cup \{\psi\}$

```
data ctxt : Type where
  ∅   : ctxt
  _::_ : ctxt → Formula → ctxt
```

```
data _∈_ : Formula → ctxt → Type where
  Z : ∀ {Γ φ} → φ ∈ Γ :: φ
  S : ∀ {Γ φ ψ} → φ ∈ Γ → φ ∈ Γ :: ψ
```

Formalizing propositional logic

Choosing logical connectives

$$\top \ \bot \ \wedge \ \vee \ \neg$$

Formalizing propositional logic

- Provability as a datatype

```
data _⊢_ : ctxt → Formula → Type where
  -- Inference rules
```

- Inference rules are inhabitants of the provability type

Formalizing propositional logic

$$\frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi} \wedge\text{-I}$$

$\wedge$-I : $\{\Gamma : \mathsf{ctxt}\}$ $\{\phi\ \psi : \mathsf{Formula}\}$
→ $\Gamma \vdash \phi$
→ $\Gamma \vdash \psi$
→ $\Gamma \vdash \phi \wedge \psi$

$$\frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi} \wedge\text{-E}_1$$

$\wedge$-E_1 : $\{\Gamma : \mathsf{ctxt}\}$ $\{\phi\ \psi : \mathsf{Formula}\}$
→ $\Gamma \vdash \phi \wedge \psi$
→ $\Gamma \vdash \phi$

$$\frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi} \wedge\text{-E}_2$$

$\wedge$-E_2 : $\{\Gamma : \mathsf{ctxt}\}$ $\{\phi\ \psi : \mathsf{Formula}\}$
→ $\Gamma \vdash \phi \wedge \psi$
→ $\Gamma \vdash \psi$

Formalizing propositional logic

Law of excluded middle

The law of excluded middle states that for every proposition, either the proposition or its negation is true.

That is, for all formulas ϕ,

$$\vdash \phi \vee \neg\phi$$

This makes it a classical logic.
In Agda:

$$\mathsf{LEM} : \{\phi : \mathsf{Formula}\} \to \emptyset \vdash \phi \vee \neg\ \phi$$

Formalizing propositional logic

There are three common structural rules:

- Weakening $$\frac{\Gamma \vdash \phi}{\Gamma, \psi \vdash \phi}$$
- Exchange $$\frac{\Gamma, \phi, \psi \vdash \gamma}{\Gamma, \psi, \phi \vdash \gamma}$$
- Contraction $$\frac{\Gamma, \phi, \phi \vdash \phi}{\Gamma, \phi \vdash \phi}$$

Only weakening and exchange are needed.

Biprovability relation

Definition

$\phi \sim \psi$ if and only if $\Gamma, \phi \vdash \psi$ and $\Gamma, \psi \vdash \phi$.

The relation is defined as a pair, so we define it as a product in Agda:

$$\_\sim\_ : \mathsf{Formula} \to \mathsf{Formula} \to \mathsf{Type}$$
$$\phi \sim \psi = (\Gamma :: \phi \vdash \psi) \times (\Gamma :: \psi \vdash \phi)$$

This is an equivalence relation!

- Reflexivity and symmetry are trivial
- Transitivity is a bit more involved...

Biprovability relation

Lemma (cut)

Given $\Gamma, \phi \vdash \gamma$ and $\Gamma, \gamma \vdash \psi$, it follows that $\Gamma, \phi \vdash \psi$.

Proof.

Natural deduction

$$\dfrac{\dfrac{\Gamma, \phi \vdash \psi}{\Gamma, \phi \vdash \psi \vee \gamma}\ \vee\text{-I}_2 \qquad \dfrac{\dfrac{\Gamma, \psi \vdash \gamma}{\Gamma, \psi, \phi \vdash \gamma}\ \text{WEAK.}}{\Gamma, \phi, \psi \vdash \gamma}\ \text{EX.} \qquad \dfrac{\gamma \in \Gamma, \phi, \gamma}{\Gamma, \phi, \gamma \vdash \gamma}\ \text{AX.}}{\Gamma, \phi \vdash \gamma}\ \vee\text{-E}$$

$\square$

Biprovability relation

Lemma (cut)

Given $\Gamma, \phi \vdash \gamma$ and $\Gamma, \gamma \vdash \psi$, it follows that $\Gamma, \phi \vdash \psi$.

Agda formalization:

```
cut : ∀ {φ ψ γ : Formula}
    → Γ :: φ ⊢ γ
    → Γ :: γ ⊢ ψ
    → Γ :: φ ⊢ ψ
cut x y = ∨-E (∨-I₂ x)
              (exchange (weakening y))
              (axiom Z)
```

Biprovability relation

Lemma

$\sim$ is an equivalence relation.

Proof.

Reflexivity and symmetry are trivial.
Transitivity: Given $\phi \sim \gamma$ and $\gamma \sim \psi$, we need to show $\phi \sim \psi$.
From the definition of $\sim$ we have

$$(1)\ \Gamma, \phi \vdash \gamma \qquad (3)\ \Gamma, \gamma \vdash \psi$$
$$(2)\ \Gamma, \gamma \vdash \phi \qquad (4)\ \Gamma, \psi \vdash \gamma$$

Using cut on (1) and (3), and on (4) and (2), we get $\Gamma, \phi \vdash \psi$ and $\Gamma, \psi \vdash \phi$, i.e $\phi \sim \psi$. $\square$

Biprovability relation

Lemma

$\sim$ is an equivalence relation.

```
∼-refl : ∀ (ϕ : Formula) → ϕ ∼ ϕ
∼-refl _ = (axiom Z , axiom Z)
```

```
∼-sym : ∀ {ϕ ψ : Formula} → ϕ ∼ ψ → ψ ∼ ϕ
∼-sym (A , B) = (B , A)
```

```
∼-trans : ∀ {ϕ ψ γ : Formula} → ϕ ∼ γ → γ ∼ ψ → ϕ ∼ ψ
∼-trans (x₁ , x₂) (y₁ , y₂) = (cut x₁ y₁ , cut y₂ x₂)
```

Table of Contents

1 Introduction and motivation

2 Agda proof assistant

3 Classical propositional logic and provability in Agda

4 Constructing the Lindenbaum-Tarski algebra using set quotioents from Cubical Agda

5 Proving the Lindenbaum-Tarski algebra is a Boolean algebra

6 Soundness

Formalizing Lindenbaum-Tarski algebra

Define Lindenbaum-Tarski algebra in Cubical Agda using the existing definition of set quotients.

$$\mathsf{LindenbaumTarski} : \mathsf{Type}$$
$$\mathsf{LindenbaumTarski} = \mathsf{Formula}\ /\ \_{\sim}\_$$

Formalizing Lindenbaum-Tarski algebra

Operations on the equivalence classes

$$\wedge/ \quad \vee/ \quad \neg/$$

To define these operations in Agda we made use of already existing definitions in the agda/cubical library. Fir example,

```
_∧/_ : LindenbaumTarski → LindenbaumTarski → LindenbaumTarski
A ∧/ B = setQuotBinOp ∼-refl ∼-refl _∧_ ∼-respects-∧ A B
```

Formalizing Lindenbaum-Tarski algebra

If $\phi \sim \phi'$ and $\psi \sim \psi'$ then $\phi \wedge \psi \sim \phi' \wedge \psi'$.

```
~-respects-∧ : ∀ (φ φ' ψ ψ' : Formula)
               → φ ~ φ'
               → ψ ~ ψ'
               → (φ ∧ ψ) ~ (φ' ∧ ψ')
~-respects-∧ φ φ' ψ ψ' (x₁ , x₂) (y₁ , y₂) =
               ∧-I (cut (∧-E₁ (axiom Z)) x₁) (cut (∧-E₂ (axiom Z)) y₁) ,
               ∧-I (cut (∧-E₁ (axiom Z)) x₂) (cut (∧-E₂ (axiom Z)) y₂)
```

Disjunction and negation are defined similarly.

Table of Contents

1 Introduction and motivation

2 Agda proof assistant

3 Classical propositional logic and provability in Agda

4 Constructing the Lindenbaum-Tarski algebra using set quotioents from Cubical Agda

5 Proving the Lindenbaum-Tarski algebra is a Boolean algebra

6 Soundness

The Lindenbaum-Tarski algebra is a Boolean algebra

- A lattice is a non-empty partially ordered set $\langle L, \leq \rangle$ where every $x, y \in L$ has a supremum $x \vee y$ and an infimum $x \wedge y$.
- A lattice L is distributive if for all $x, y, z \in L$,

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

- A lattice L is complemented if there exist both least and greatest elements in L denoted $\perp$ and $\top$, and for every $x \in L$ there exists $y \in L$ such that

$$x \vee y = \top \quad \text{and} \quad x \wedge y = \perp$$

- A Boolean algebra is a complemented distributive lattice

The Lindenbaum-Tarski algebra is a Boolean algebra

- Showing the Lindenbaum-Tarski algebra is a complemented distributive lattice implies that it is a Boolean algebra.
- There is an existing definition of a distributive lattice in the agda/cubical library.

```
LindenbaumTarski-DistLattice : DistLattice _
LindenbaumTarski-DistLattice = makeDistLattice∧lOver∨l
                                 ⊥/ ⊤/ _∨/_ _∧/_
                                 isSet-LT
                                 ∨/-ass ∨/-id ∨/-comm
                                 ∧/-ass ∧/-id ∧/-comm ∧/-abs ∧/-dist
```

The Lindenbaum-Tarski algebra is a Boolean algebra

So far we have shown that it is a distributive lattice. Now we must show that it is also complemented. This follows from *Law of excluded middle* and *Law of non-contraction*, properties that are present in the propositional logic.

```
LindenbaumTarski-DistLattice-supremum :
  (A : fst LindenbaumTarski-DistLattice)
  → A ∨l ¬/ A ≡ 1l
LindenbaumTarski-DistLattice-supremum A = ∨/-comp A
```

```
LindenbaumTarski-DistLattice-infimum :
  (A : fst LindenbaumTarski-DistLattice)
  → A ∧l ¬/ A ≡ 0l
LindenbaumTarski-DistLattice-infimum A = ∧/-comp A
```

Table of Contents

1 Introduction and motivation

2 Agda proof assistant

3 Classical propositional logic and provability in Agda

4 Constructing the Lindenbaum-Tarski algebra using set quotioents from Cubical Agda

5 Proving the Lindenbaum-Tarski algebra is a Boolean algebra

6 Soundness

Soundness

- With the Lindenbaum-Tarski algebra formalized, we can use it to prove properties about propositional logic.
- If $\vdash \phi$ then $[\phi] \equiv [\top]$. We can view this as a form of soundness.

```
sound : ∀ {ϕ : Formula} → ∅ ⊢ ϕ → [ ϕ ] ≡ ⊤/
sound x = eq/ _ _ (superweakening ⊤-I , superweakening x)
```

```
superweakening : ∀ {Γ : ctxt} {ϕ : Formula} → ∅ ⊢ ϕ → Γ ⊢ ϕ
superweakening {∅} x = x
superweakening {Δ :: ψ} x = weakening (superweakening x)
```

Discussion/conclusion

#TODO

- Structural rules
- Implication?