Pre-bilattices in Univalent Foundations

Georgios V. Pitsiladis
gpitsiladis@mail.ntua.gr

School of Applied Mathematical and Physics Science,
National Technical University of Athens, Greece

TYPES 2022
21 June 2022

Univalent Foundations and UniMath

Univalent Foundations is the pursue to formalise mathematics using Homotopy Type Theory.

UniMath is a library for Univalent Foundations built on the Coq theorem prover.

It is thus a constructive structural framework in which it is possible to formalise mathematical entities and proofs.
Some algebraic structures have already been formalised in UniMath.

Motivation

Explore how bilattices and their basic theory can be defined in UniMath.

Work in progress.
The (not yet polished) code can be found in

`https://github.com/giorgio93p/UniMath`,

branch `bilattices`, file `UniMath/Algebra/Bilattice.v`.

Lattices

Lattices are partially ordered sets where every pair of elements has an infimum (greatest lower bound) and a supremum (least upper bound).

or, equivalently,

Lattices are sets with two associative and commutative binary operators linked by absorption properties.

UniMath contains a formalisation of lattices.

$\texttt{lattice : hSet} \rightarrow \mathcal{U}$.

Definition of pre-bilattices

Pre-bilattices are sets equipped with two lattice structures:

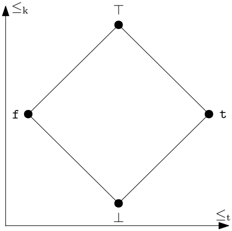
```
Definition prebilattice (X : hSet) :=
    lattice X × lattice X .
```

The 1st lattice is called the *truth* lattice (`tlattice`);
the 2nd lattice is called the *knowledge* lattice (`klattice`).

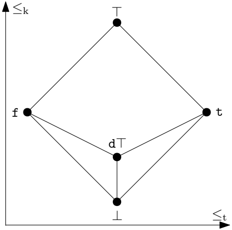
Terminology:

lattice	order	inf/glb	sup/lub
truth	`tle`	meet	join
knowledge	`kle`	consensus	gullibility

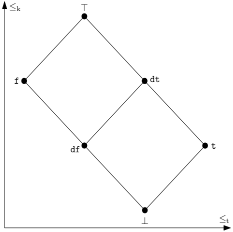
Examples of pre-bilattices



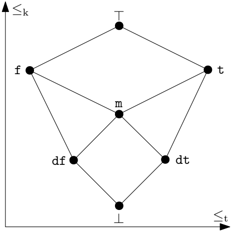
(a) $\mathcal{FOUR}$



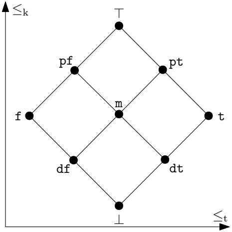
(b) $\mathcal{FIVE}$



(c) $\mathcal{SIX}$



(d) $\mathcal{DEFAULT}$



(e) $\mathcal{NINE}$

Figure: Some of the most common pre-bilattices.

Dualities in pre-bilattices

Observe that the definition is symmetric, in the sense that we can swap the two lattices and still obtain a pre-bilattice. This gives rise to a duality.
Dualities are ideal for proof reuse.

Dualities in pre-bilattices

Observe that the definition is symmetric, in the sense that we can swap the two lattices and still obtain a pre-bilattice. This gives rise to a duality.
Dualities are ideal for proof reuse. For example,

```
Lemma property1 {X : hSet} :
     ∏ (b : interlaced_prebilattice X) (x y : X) ,
          (∑r : X, tle b x r × kle b r y)
              → kle b x (meet b y x).

Lemma property1_dual {X : hSet} :
    ∏ (b : interlaced_prebilattice X) (x y : X) ,
         (∑r : X, kle b x r × tle b r y)
             → tle b x (consensus b y x).
```

Dualities in pre-bilattices

Observe that the definition is symmetric, in the sense that we can swap the two lattices and still obtain a pre-bilattice. This gives rise to a duality.
Dualities are ideal for proof reuse. For example,

```
Lemma property1 {X : hSet} :
     ∏ (b : interlaced_prebilattice X) (x y : X) ,
         (∑r : X, tle b x r × kle b r y)
            → kle b x (meet b y x).

Lemma property1_dual {X : hSet} :
    ∏ (b : interlaced_prebilattice X) (x y : X) ,
        (∑r : X, kle b x r × tle b r y)
           → tle b x (consensus b y x).
Proof.
  intro b.
  use (property1 (make_interlaced_prebilattice (
     dual_prebilattice_is_interlaced b))).
Defined.
```

Dualities in pre-bilattices

Observe that the definition is symmetric, in the sense that we can swap the two lattices and still obtain a pre-bilattice. This gives rise to a duality.
Dualities are ideal for proof reuse. For example,

```
Lemma property1 {X : hSet} :
     ∏ (b : interlaced_prebilattice X) (x y : X) ,
          (∑r : X, tle b x r × kle b r y)
               → kle b x (meet b y x).

Lemma property1_dual {X : hSet} :
     ∏ (b : interlaced_prebilattice X) (x y : X) ,
          (∑r : X, kle b x r × tle b r y)
               → tle b x (consensus b y x).
Proof.
  intro b.
  use (property1 (make_interlaced_prebilattice (
       dual_prebilattice_is_interlaced b))).
Defined.
```

Moreover, each of the two lattices can be inverted and still obtain a pre-bilattice. This gives rise to more dualities.

Interlaced and distributive pre-bilattices

A pre-bilattice is called *interlaced* iff each of meet, join, consensus, gullibility is monotone w.r.t. both orders.

Interlaced and distributive pre-bilattices

A pre-bilattice is called *interlaced* iff each of meet, join, consensus, gullibility is monotone w.r.t. both orders.

A pre-bilattice is called *distributive* if all pairs of meet, join, consensus, gullibility obey the distributive property.

Interlaced and distributive pre-bilattices

A pre-bilattice is called *interlaced* iff each of meet, join, consensus, gullibility is monotone w.r.t. both orders.

A pre-bilattice is called *distributive* if all pairs of meet, join, consensus, gullibility obey the distributive property.

It is simple to prove that

```
Theorem distributive_prebilattices_are_interlaced
    {X : hSet} (b : distributive_prebilattice X)
     : is_interlaced b .
```

Product pre-bilattices

Given `l1 : lattice X1`, `l2 : lattice X2`, we can construct a pre-bilattice (`prod_prebilattice X1 X2 l1 l2`) over `X1 × X2`. Its meet, join, consensus, and gullibility are defined by:

meet	(inf_1, sup_2)
join	(sup_1, inf_2)
consensus	(inf_1, inf_2)
gullibility	(sup_1, inf_2)

The representation theorem

Theorem

1. *Every product pre-bilattice is interlaced.*
2. *Every interlaced pre-bilattice is isomorphic to a product pre-bilattice.*

The first item can be proved using the definition of product pre-bilattices and (many) properties of lattices.

The second item is more tricky.

The representation theorem: interlaced $\rightarrow$ product

A proof is presented in: F. Bou and U. Rivieccio. The logic of distributive bilattices. *Logic Journal of IGPL*, 19(1):183–216.

- Define two specific equivalence relations R_1, R_2 on the elements of the underlying set X of the interlaced pre-bilattice.

  ```
  Definition R1 {X : hSet}
      (b : interlaced_prebilattice X) : hrel X :=
    λ x y : X, eqset (join b x y) (consensus b x y)   .
  ```

  R_2 is t-dual to R_1.
- Form a lattice L_1 using the set of equivalence classes $[X]_{R_1}$ with `gullibility b` as sup and `consensus b` as inf.
- Form a lattice L_2 using the set of equivalence classes $[X]_{R_2}$ with `gullibility b` as sup and `consensus b` as inf.
- The interlaced pre-bilattice is equivalent to the product pre-bilattice of L_1 and L_2.

Proving the representation theorem (interlaced → product) in UniMath

It is necessary to use `setquotunivprop`

1. to establish the compatibility of R_1 and R_2 with pre-bilattice operators,
2. to prove the equivalence of the underlying sets of the initial interlaced pre-bilattice and the resulting product pre-bilattice.

We need to move from equivalence of sets to equivalence of pre-bilattices. However, `setquotunivprop` is opaque. As mentioned in UniMath, "Terms produced using [`setquotunivprop`] will not fully normalize even in simple cases." Hence, proving properties of terms including it is not straightforward.

Proving the representation theorem (interlaced $\to$ product) in UniMath

It is necessary to use `setquotunivprop`

1. to establish the compatibility of R_1 and R_2 with pre-bilattice operators,
2. to prove the equivalence of the underlying sets of the initial interlaced pre-bilattice and the resulting product pre-bilattice.

We need to move from equivalence of sets to equivalence of pre-bilattices. However, `setquotunivprop` is opaque. As mentioned in UniMath, "Terms produced using [`setquotunivprop`] will not fully normalize even in simple cases." Hence, proving properties of terms including it is not straightforward.

Solution: Structure Identity Principle (SIP)

Proving the representation theorem (interlaced → product) in UniMath: SIP

1. Build the precategory of (hSet, pre-bilattice) pairs.
   Morphisms are set functions that respect pre-bilattice structure.
2. SIP states that isomorphisms in this category are equalities.
3. Hence, by proving that the interlaced → product construction respects pre-bilattice structure, we can obtain

```
eq_interlaced_product :
  ∏ (X : hSet) (b : interlaced_prebilattice X),
  ∑ (X1 X2 : hSet)
    (l1 : lattice X1) (l2 : lattice X2)
    (b' : prod_prebilattice X1 X2 l1 l2),
      X,, b = X1 × X2,, b'
```

Future extensions

The representation theorem and the many dualities of pre-bilattices can be used to prove properties of (interlaced) pre-bilattices.

The representation theorem has a version for distributive pre-bilattices, stating that a product pre-bilattice is distributive iff the two lattices that were used to construct it are distributive.

Pre-bilattices that are symmetric w.r.t. truth order support negation.
Dually, pre-bilattices that are symmetric w.r.t. knowledge order support conflation.

Pre-bilattices with at least one of the above operators are called *bilattices*.

The representation theorem (and the notion of product pre-bilattices) has a version for bilattices, taking the extra operators into account.

Thank you!