

# Structurer les mathématiques, des primitives aux fonctionnalités

## Les mécanismes dédiés

### Primitives

- Coercions.
- Structures canoniques.
- Typeclasses.
- Coercions inverses.

### Limites

- Coercions: pas de "default class": pas de coercion implicite d'un sous-type vers le type de base. (En lean réalisé par les typeclass)
- Structures canoniques : une seule solution canonique, force au bundling
- Typeclasses : force le unbundling.

Aucun de ces mécanismes n'est descriptif : il est nécessaire d'établir des design patterns.

## Design pattern et Hierarchy Builder

Voir mathcomp #733

## Traitement des morphismes et sous-objets

- fonctions  $f : \{\text{rmorphism } R \rightarrow S\}$
- sous-objets  $r : \{\text{subring } R\}$

Liens entre sous-typage et sous-objets

3 niveaux de stabilité :

1. stabilité des types si  $T$  et  $T'$  ont une structure alors  $T + T'$  aussi
2. stabilité des ensembles si  $A$  et  $B$  (de type  $\text{set } T$ ) ont une sous-structure de  $T$ , alors  $A + B$  aussi.
3. stabilité des éléments si  $x \in A$  et  $y \in A$ , alors  $x + y \in A$

Questions supplémentaires: opérations hétérogènes :

$a : A$  et  $b : B$  avec  $A \leq B$ , quid de  $a + b$ ?