Relational ornaments

Josh Ko & Jeremy Gibbons

Department of Computer Science
University of Oxford

Workshop on Dependently Typed Programming
Vienna, Austria, 13 July 2014

Ornamentation

```
data List (A : Set) : Set where
  List A ∋ nil
         | cons (x : A) (xs : List A)
```

```
data Vec (A : Set) : Nat → Set where
  Vec A zero     ∋ nil
  Vec A (suc n) ∋ cons (x : A) (xs : Vec A n)
```

Ornamentation

```
data List (A : Set) : Set where
  List A ∋ nil
         | cons (x : A) (xs : List A)
```

```
data Vec (A : Set) : Nat → Set where
  Vec A zero     ∋ nil
                 | cons
  Vec A (suc n) ∋ nil
                 | cons (x : A) (xs : Vec A n)
```

Descriptions

```
data Desc (I : Set) : Set₁
   ListD : Set → Desc ⊤
data μ (D : Desc I) : I → Set
List A = μ (ListD A) tt : Set
```

Ornaments

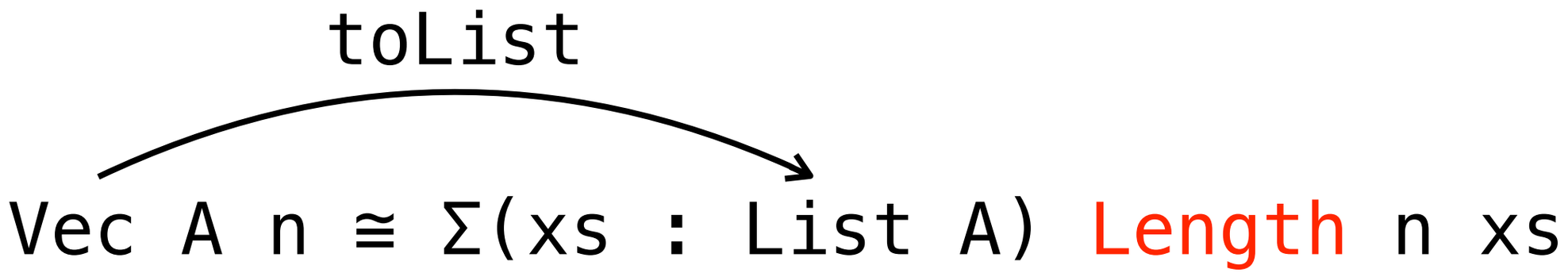
```
data Orn {I : Set} (J : Set)
         (e : J → I)
         (D : Desc I) : Set₁

   VecO : (A : Set) → Orn Nat ! (ListD A)

⌊_⌋ : Orn J e D → Desc J

   Vec A = μ ⌊ VecO A ⌋ : Nat → Set

forget : (O : Orn J e D) →
         μ ⌊ O ⌋ j → μ D (e j)

   toList = forget (VecO A): Vec A n → List A
```

Relational Ornaments

```
data Orn {I : Set} {J : Set}
         (e : J → I)
         (D : Desc I)
         (E : Desc J) : Set₁

  VecD : Set → Desc Nat

  ListD-VecD : (A : Set) →
               Orn ! (ListD A) (VecD A)

forget : (O : Orn e D E) → μ E j → μ D (e j)

  toList =
    forget (ListD-VecD A) : Vec A n → List A
```

Theory of parallel composition

Ornaments induce families of isomorphisms …



… and they can be composed.

```
OrdList b ≅ Σ(xs : List Val) Ordered b xs

OrdVec b n ≅ Σ(xs : List Val)
               Ordered b xs × Length n xs
```

Ordered vectors

```
data OrdList : Val → Set where
  OrdList b ∋ nil
            | cons (x : Val) (l : b ≤ x)
                   (xs : OrdList x)
```

```
data Vec Val : Nat → Set where
  Vec Val zero     ∋ nil
  Vec Val (suc n) ∋ cons (x : Val)
                         (xs : Vec Val n)
```

Ordered vectors

```
data OrdVec : Val → Nat → Set where
  OrdVec b zero     ∋ nil
  OrdVec b (suc n) ∋ cons (x : Val)
                          (l : b ≤ x)
                          (xs : OrdVec x n)
```

Orderedness predicate

```
data OrdList : Val → Set where
  OrdList b ∋ nil
            | cons (x : Val) (l : b ≤ x)
                   (xs : OrdList x)
```

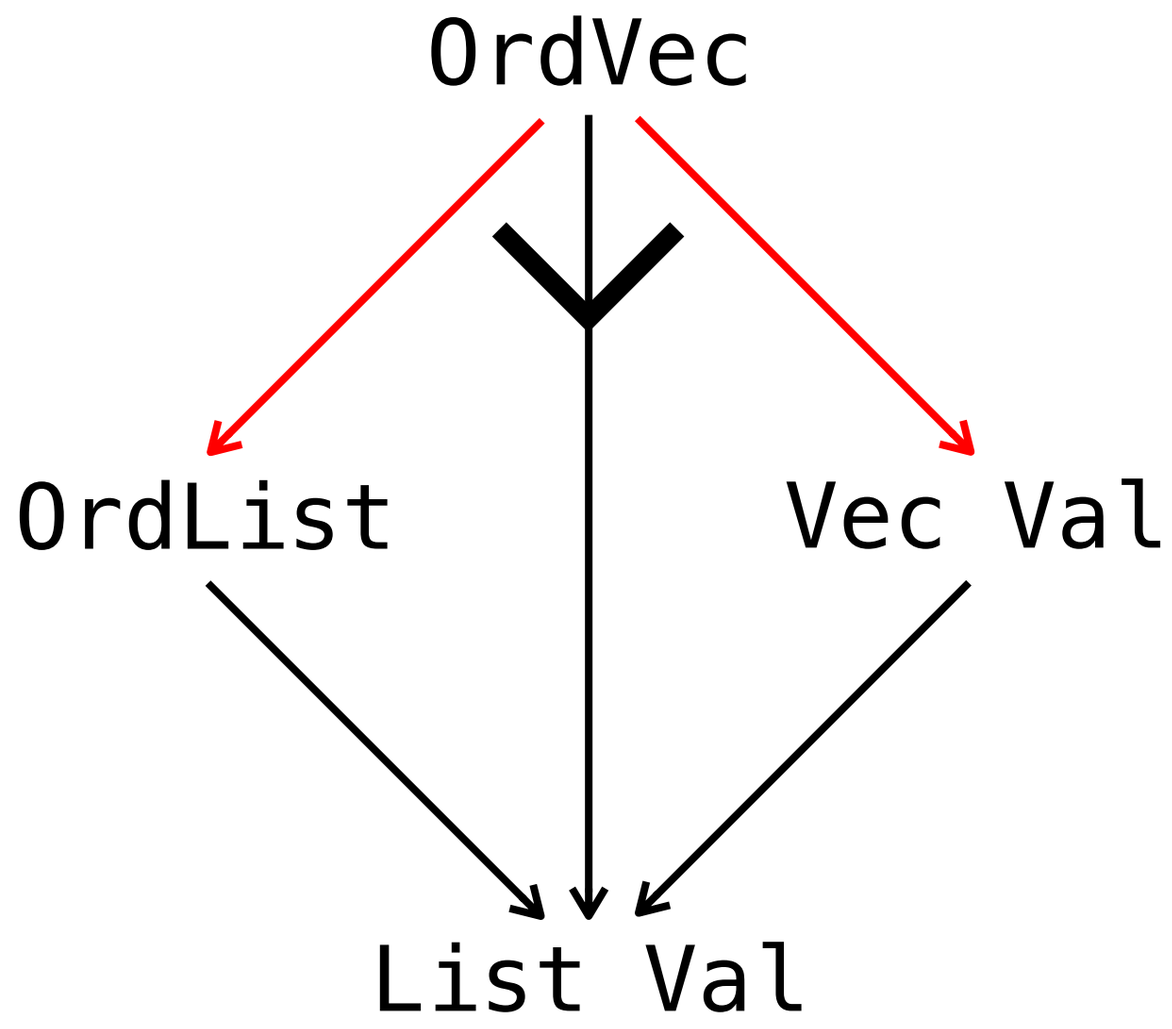
```
data ListS : List Val → Set where
  ListS nil           ∋ nil
  ListS (cons x xs) ∋ cons ~~(x : Val)~~
                           (s : ListS xs)
```

Orderedness predicate

```
data Ordered : Val → List Val → Set where
  Ordered b nil          ∋ nil
  Ordered b (cons x xs)
∋ cons (x : Val) (l : b ≤ x)
       (ls : Ordered x xs)
```

Parallel composition

is a pullback in the category of descriptions and ornaments



sequential composition

```
_⊙_ : Orn e D E → Orn f E F → Orn (e ∘ f) D F
```

From the pullback property

we can derive

```
   Vec A n ≅ Σ(xs : List A) Length n xs
OrdList b ≅ Σ(xs : List Val) Ordered b xs
```

```
OrdVec b n ≅ Σ(xs : List Val)
               OrderedLength b n xs
           ≅ Σ(xs : List Val)
               Ordered b xs × Length n xs
```

whose constructions are independent of choices of universe encodings.

Datatypes à la carte

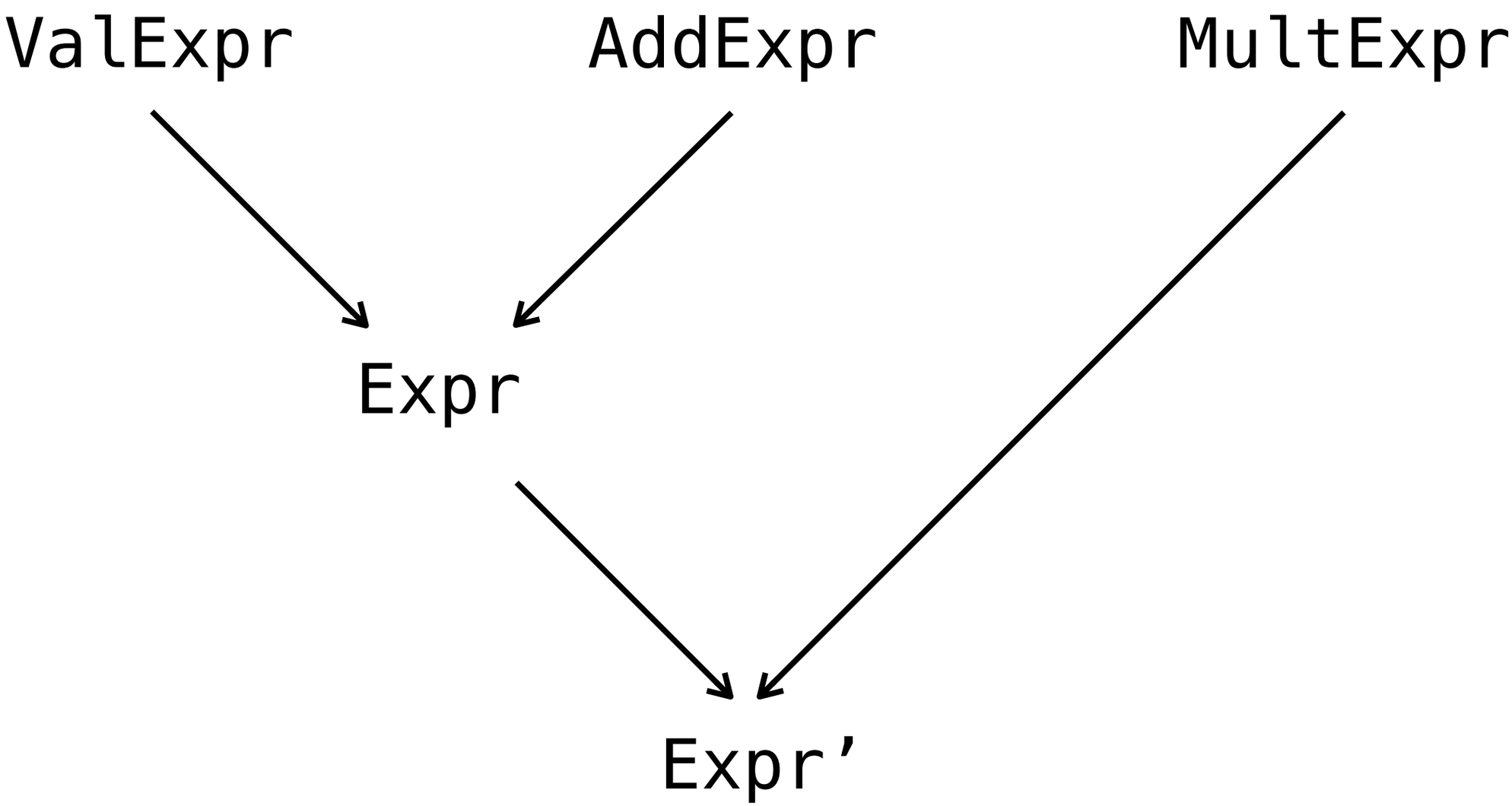
Coproduct of signatures

```
data ValExpr : Set where
  ValExpr ∋ val (n : Nat)
          | add
```

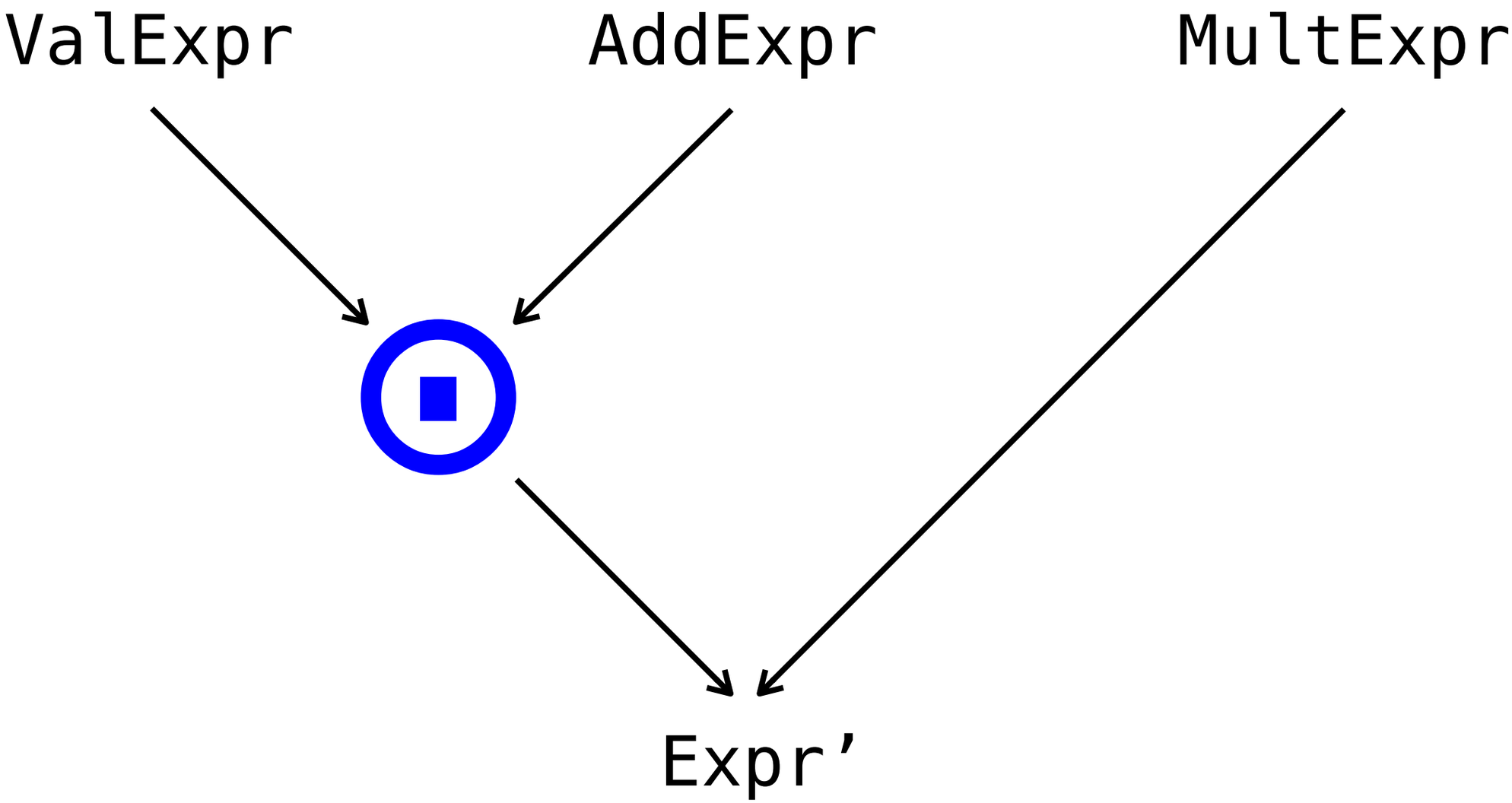
```
data AddExpr : Set where
  AddExpr ∋ val
          | add (l : AddExpr)
                (r : AddExpr)
```

```
data Expr : Set where
  Expr ∋ val (n : Nat)
       | add (l : Expr) (r : Expr)
```

Even more signatures



Even more signatures



Smart constructors

```
AddExprD : Desc ⊤
  _+_ : {{_ : Orn ! D AddExprD}} →
        μ D tt → μ D tt → μ D tt
```

Summary

With relational ornaments, creation of datatypes is decoupled from specification of ornamental relationship between datatypes.

- Ornaments can then arise between existing datatypes,
  - leading to a nice theory of parallel composition.
- New datatypes can arise at the less informative end,
  - so *datatypes à la carte* may be subsumed in the theory of ornaments.