Commuting Datatypes

Roland Backhouse and Paul Hoogendijk

Generic Programming Summer School

Oxford, August 2002

Introductory Examples

Zip (of lists)

$$([a_1, a_2, \ldots, a_n], [b_1, b_2, \ldots, b_n]) \mapsto [(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)]$$

$$\mathsf{Pair} \cdot \mathsf{List} \quad \mapsto \quad \mathsf{List} \cdot \mathsf{Pair}$$

Matrix Transposition

$$\mathsf{List} \cdot \mathsf{List} \quad \mapsto \quad \mathsf{List} \cdot \mathsf{List}$$

Broadcast

$$(a, [b_1, b_2, \ldots, b_n]) \mapsto [(a, b_1), (a, b_2), \ldots, (a, b_n)]$$

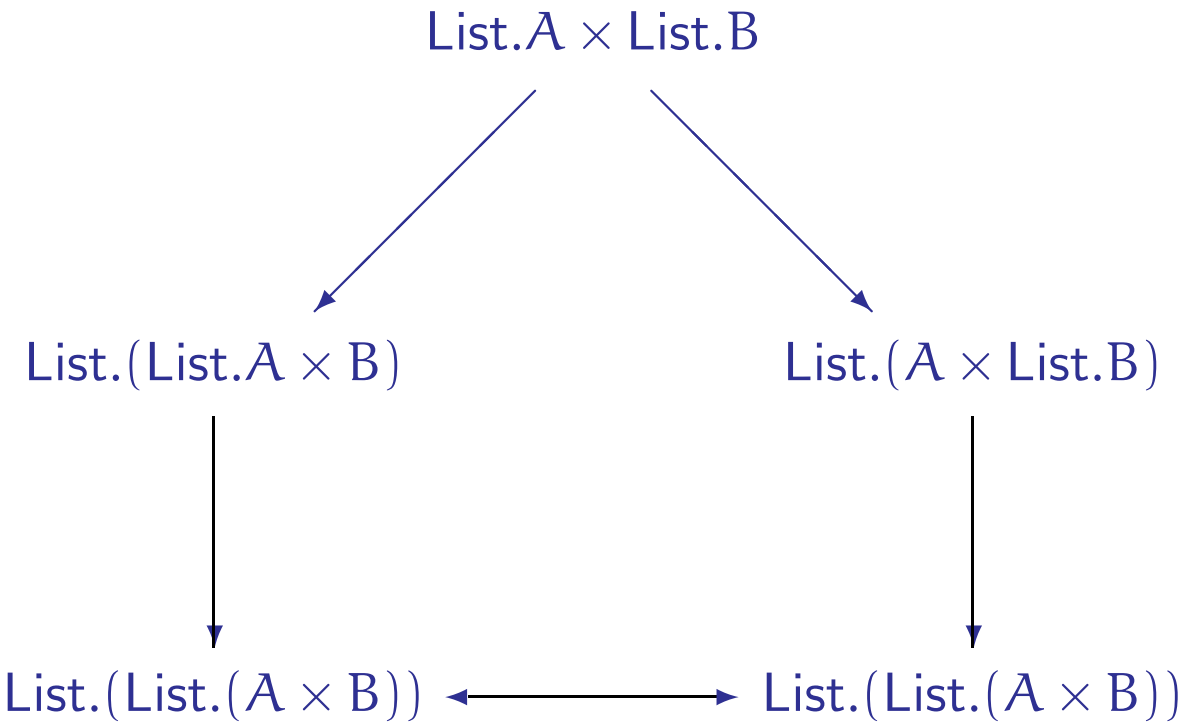
$$A\times \cdot \mathsf{List} \quad \mapsto \quad \mathsf{List} \cdot A\times$$

Primitive

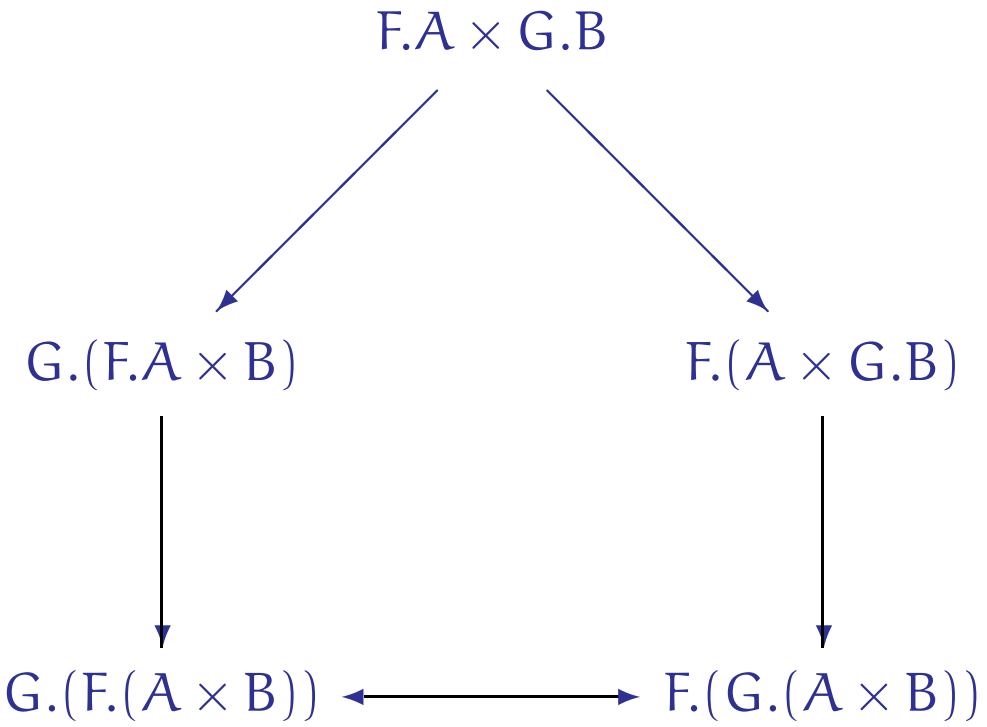
$$(A+B)\times(C+D) \mapsto (A\times C)+(B\times D)$$

$$\times \cdot + \quad \mapsto \quad + \cdot \times$$

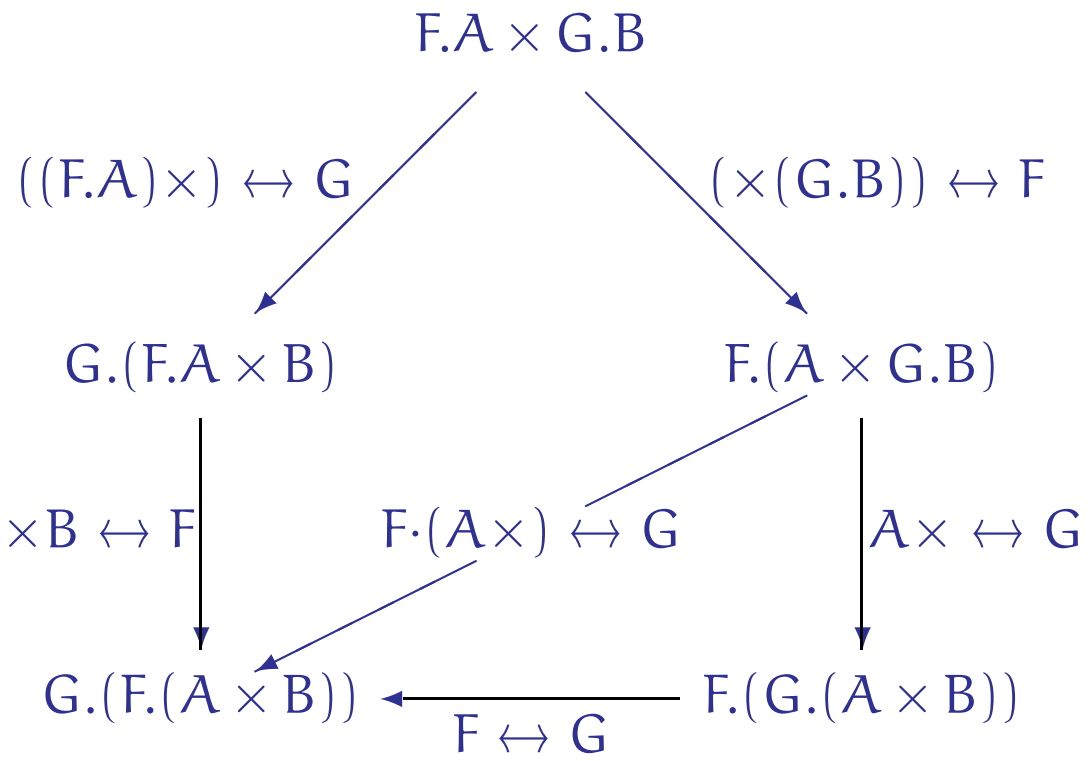
Structure Multiplication ...



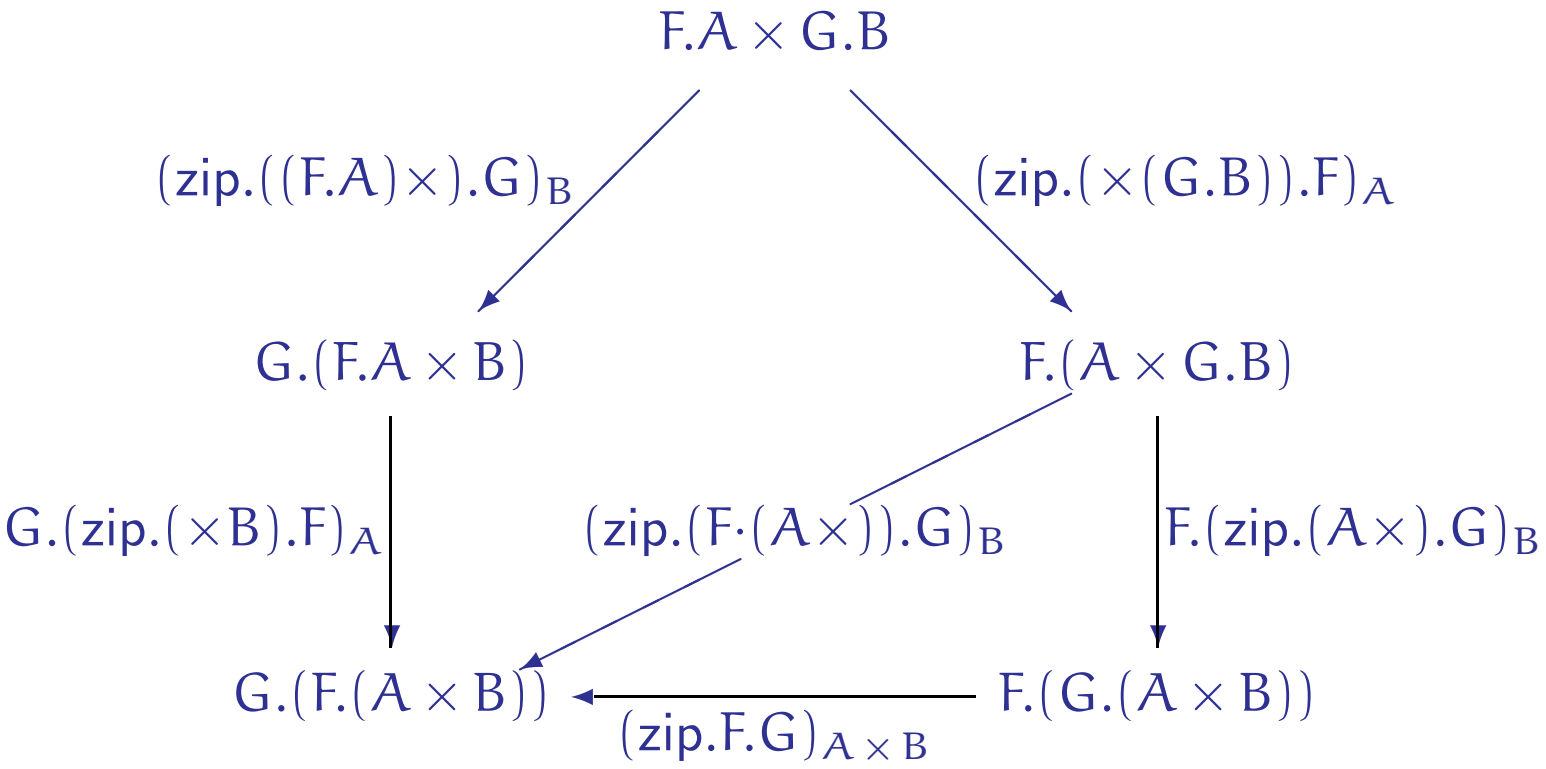
... Generalised ...



... Illustrates Generic Requirements



Multi-Coloured Zips



Broadcasts ...

A broadcast copies a given value across all storage locations of a datatype.

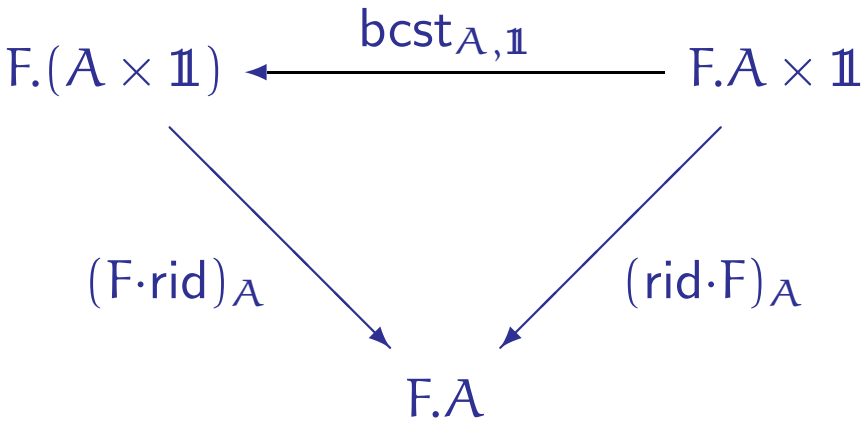
Formally, a family of functions bcst, where

$$\mathsf{bcst}_{A,B} \;:\; \mathsf{F}.(A \times B) \leftarrow \mathsf{F}.A \times B$$

is said to be a *broadcast* for datatype F iff it is parametrically polymorphic in the parameters A and B and $\mathsf{bcst}_{A,B}$ behaves coherently with respect to product in the following sense:

... Respect the Unit of Product ...

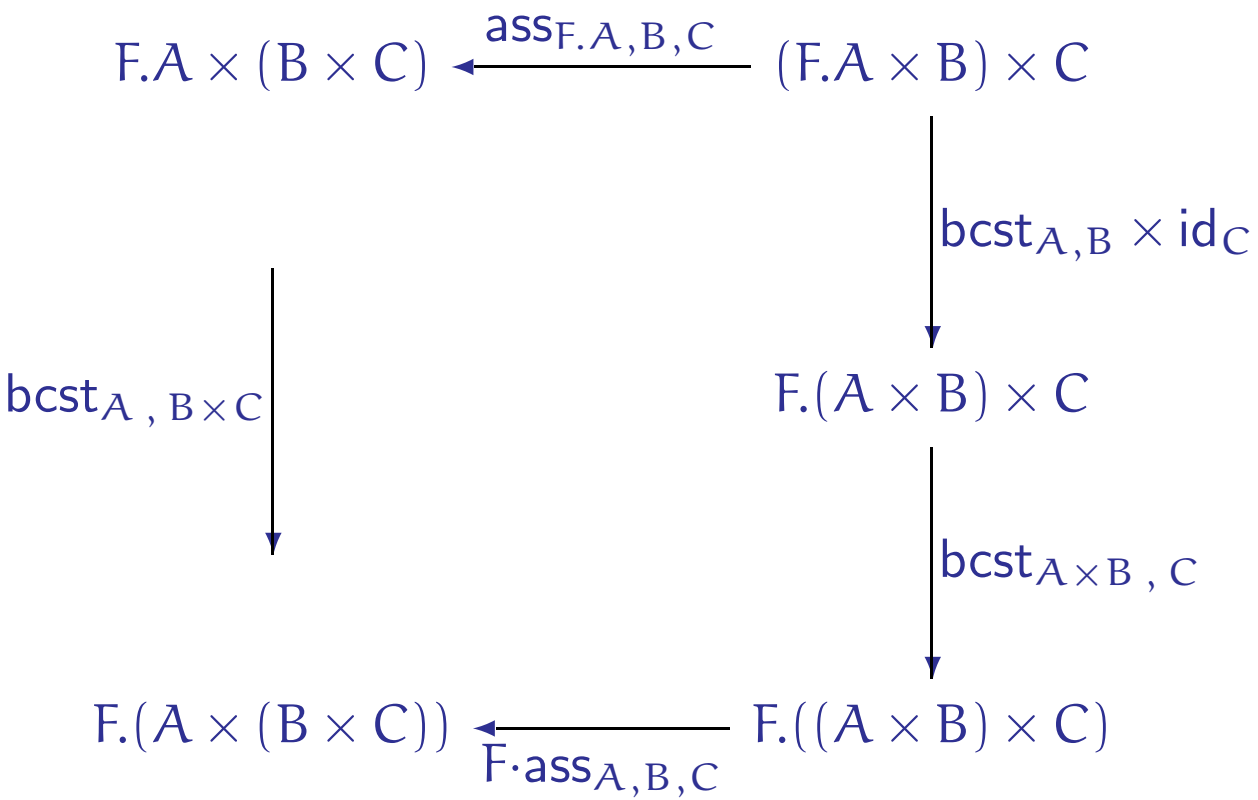
The following diagram



(where $\mathsf{rid}_A : A \leftarrow A \times \mathbb{1}$ is the obvious natural isomorphism) commutes.

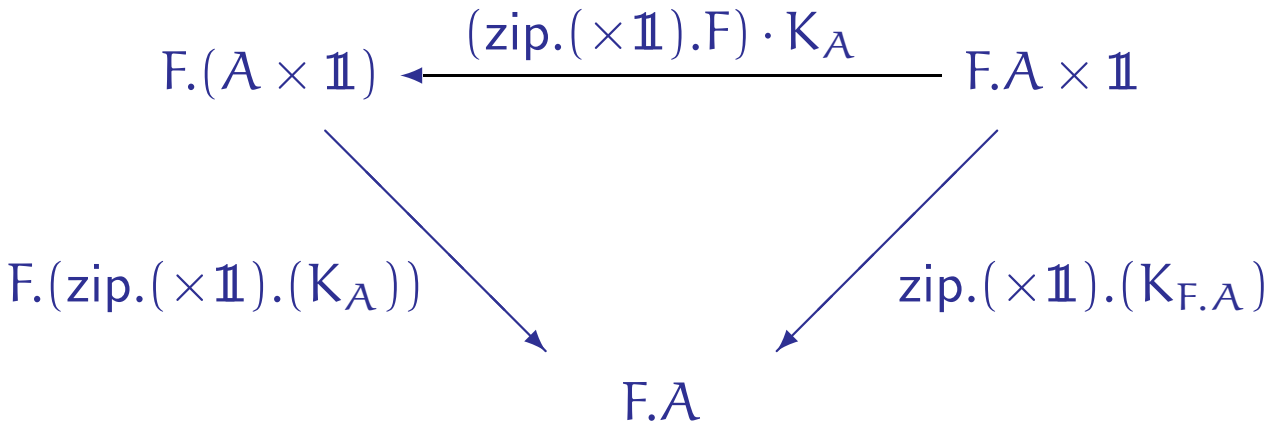
... and Associativity of Product

The following diagram

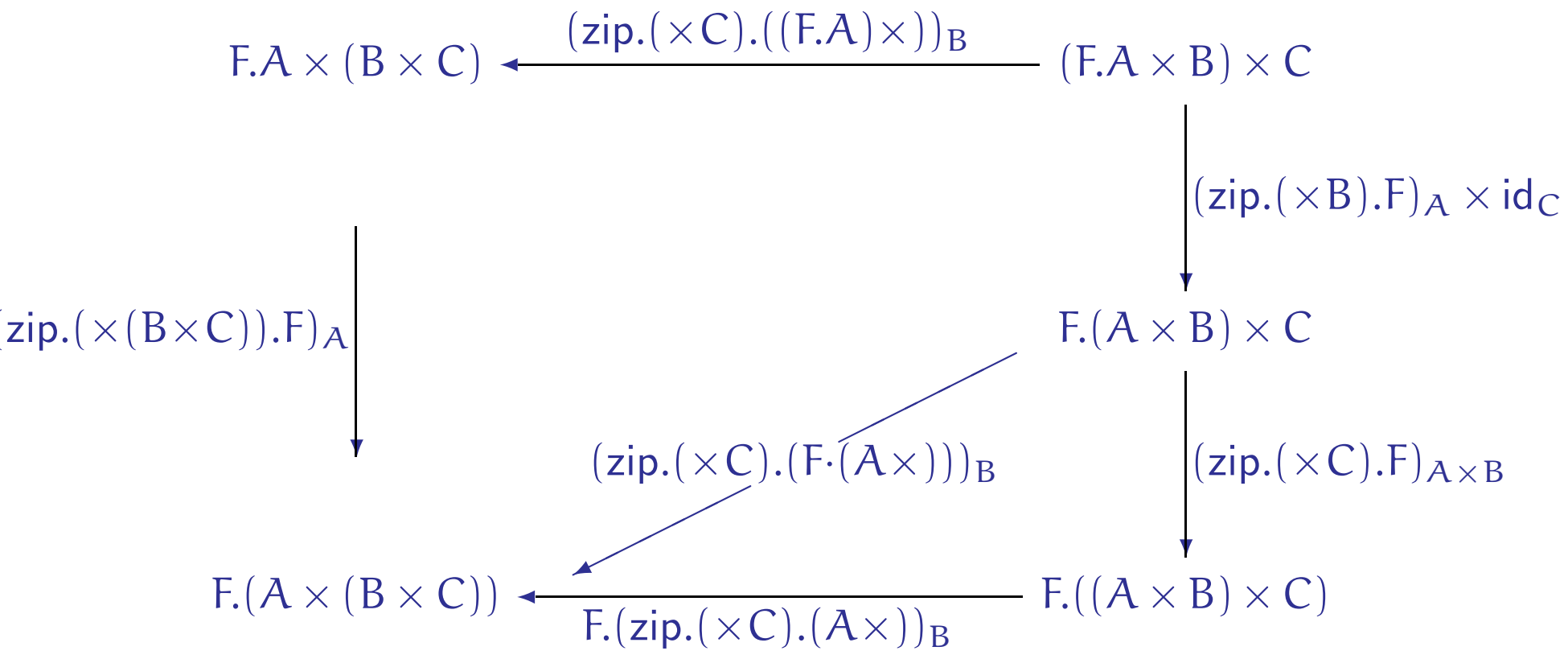


(where $\mathsf{ass}_{A,B,C} : A \times (B \times C) \leftarrow (A \times B) \times C$ is the obvious natural isomorphism) commutes as well.

Unit of Product is a "zip"



Associativity of Product is a "Zip"



Conclusion

- Commuting Datatypes ("Zips") are everywhere!
- Generic specification and proof is (potentially) very effective.
- A relational framework is necessary.
- Challenge: give generic specification of "commuting datatypes" from which "zips" can be constructed calculationally.