Dragging Proofs out of Pictures

Ralf Hinze [1] Dan Marsden [2]

[1]Institute for Computing and Information Sciences
Radboud University, Nijmegen, The Netherlands
ralf@cs.ru.nl

[2]Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, Oxford, OX1 3QD, England
daniel.marsden@cs.ox.ac.uk

April 2016

1 Proofs in Elementary Category Theory

- calculational reasoning
  - *squiggolling*
- commuting diagrams
  - *diagram chasing*
- best of both worlds: string diagrams
  - *lollipops, forks, . . .*
  - *weaving, dragging, sliding, . . .*

1 Proofs in Elementary Category Theory

- calculational reasoning
  - *squiggolling*
- commuting diagrams
  - *diagram chasing*
- best of both worlds: string diagrams
  - *lollipops, forks, . . .*
  - *weaving, dragging, sliding, . . .*

2 Monads: Algebraic Definition

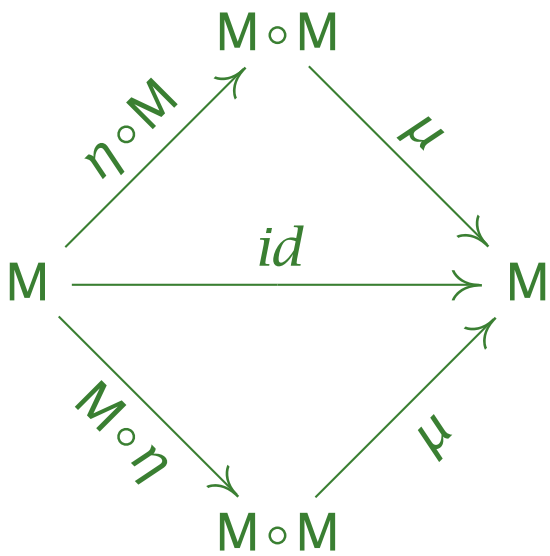
$$\mathsf{M} : C \to C$$
$$\eta : \mathsf{Id} \stackrel{\cdot}{\to} \mathsf{M}$$
$$\mu : \mathsf{M} \circ \mathsf{M} \stackrel{\cdot}{\to} \mathsf{M}$$

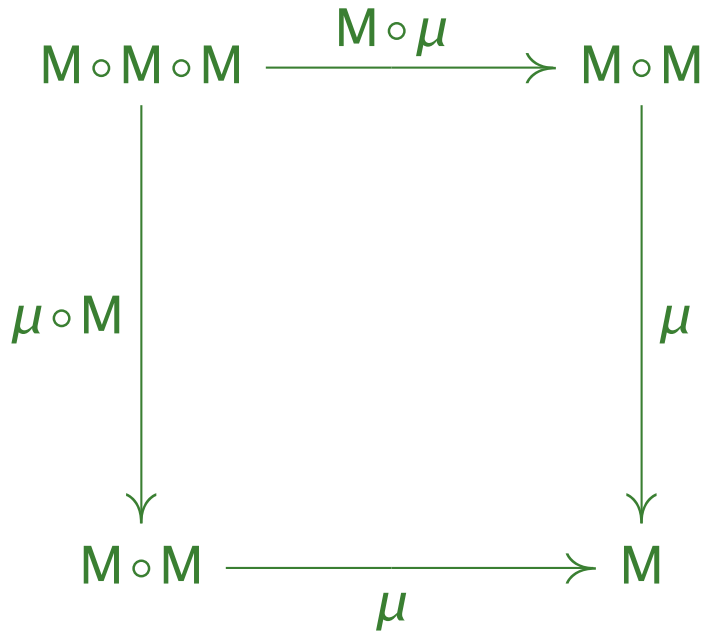
$$\mu \cdot (\eta \circ \mathsf{M}) = id = \mu \cdot (\mathsf{M} \circ \eta)$$
$$\mu \cdot (\mu \circ \mathsf{M}) = \mu \cdot (\mathsf{M} \circ \mu)$$

2 Monads: "Essence of FP"

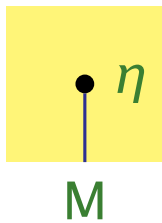
```
mapM id      =  id
mapM (f.g)   =  mapM f . mapM g

mapM f . unitM   =  unitM . f
mapM f . joinM   =  joinM . mapM (mapM f)

joinM . unitM         =  id
joinM . mapM unitM    =  id
joinM . mapM joinM    =  joinM . joinM

m `bindM` k   =  joinM (mapM k m)
```

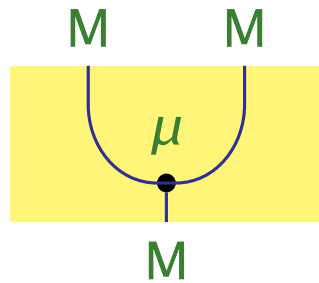
2 Monads: Commuting Diagrams



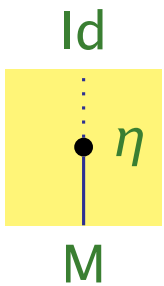


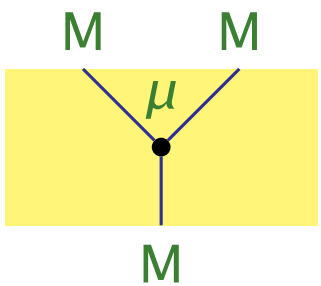
2 Monads: String Diagrams



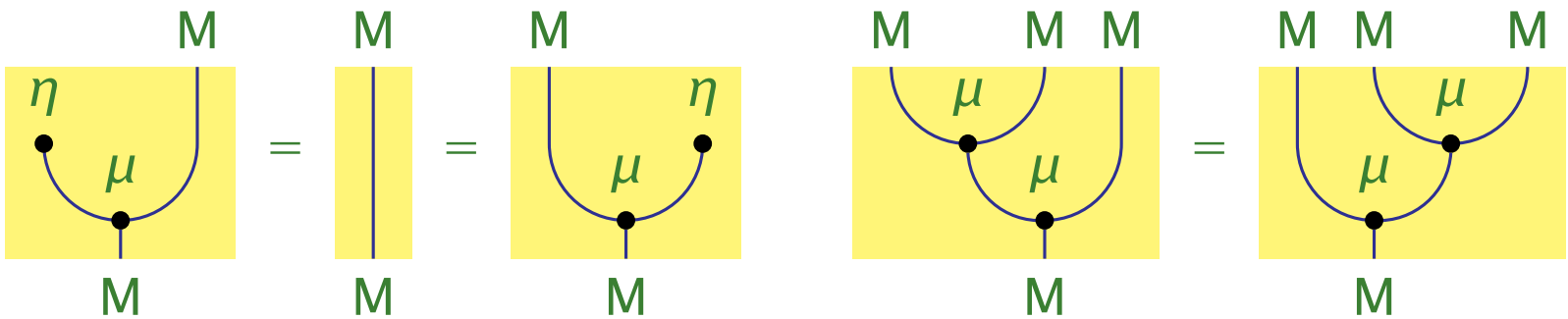


2 Monads: String Diagrams

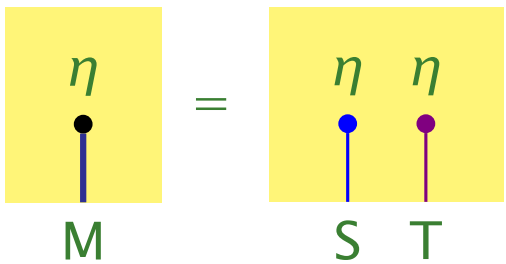




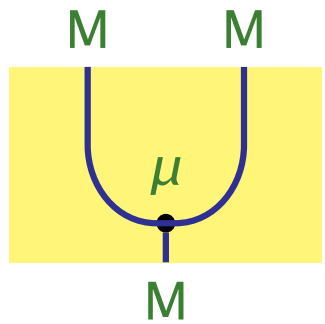
2 Monads: String Diagrams



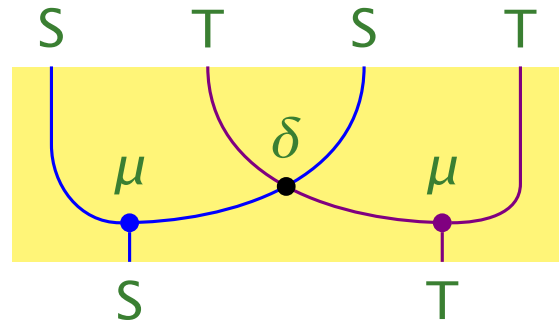
3 Composition of Monads



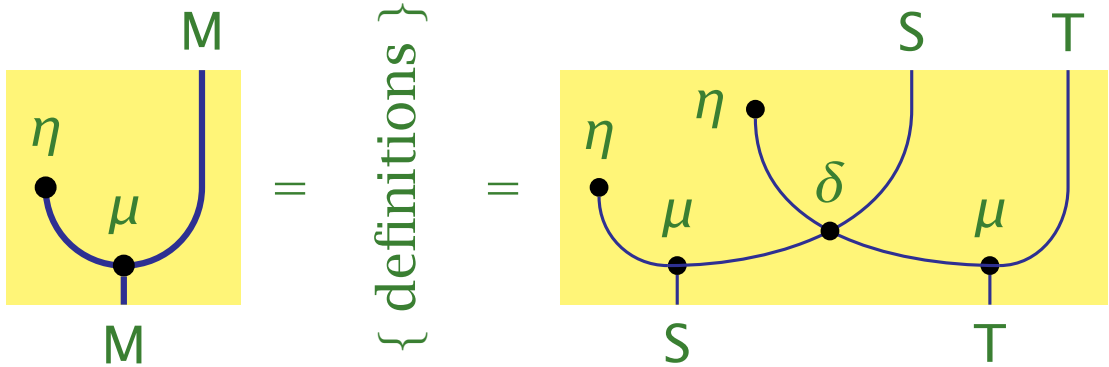
=



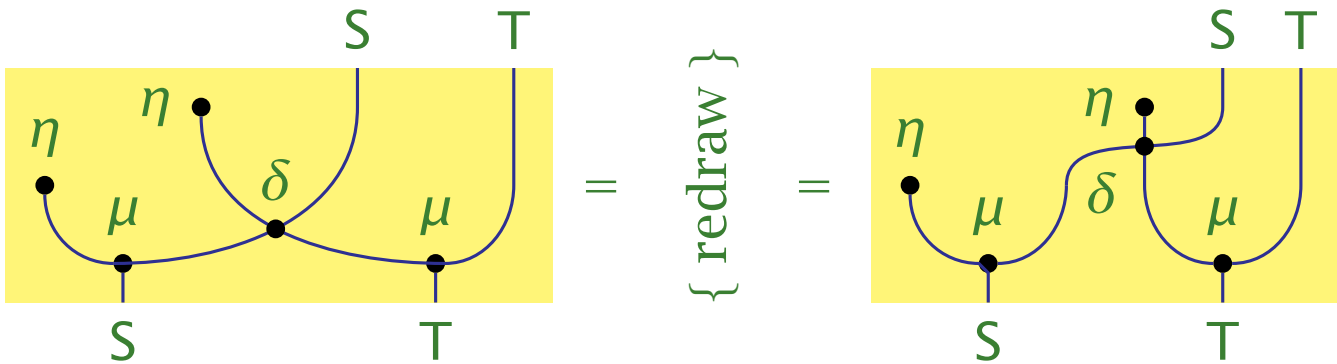
=



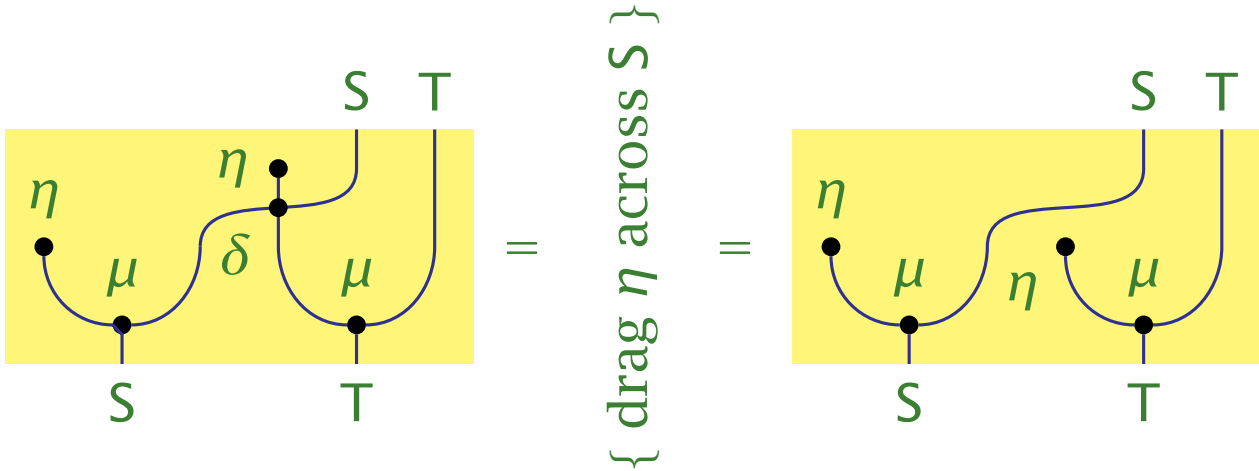
3 Composition of Monads: Unit Law



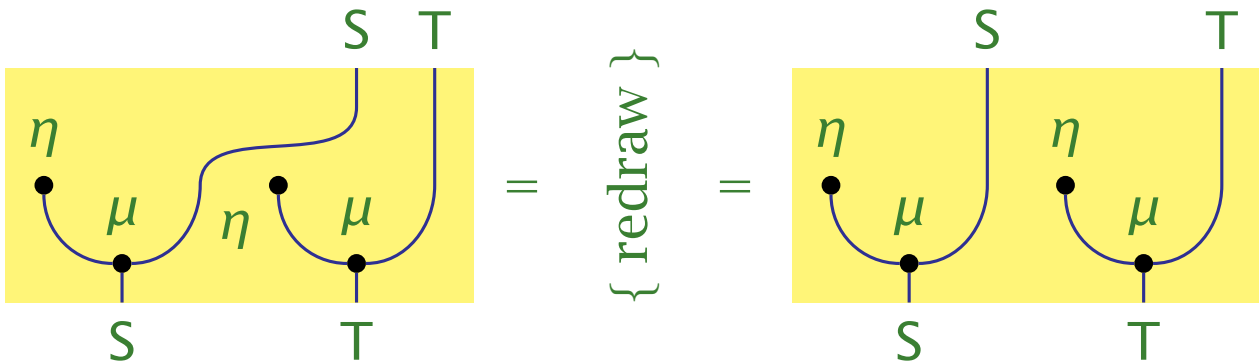
3 Composition of Monads: Unit Law



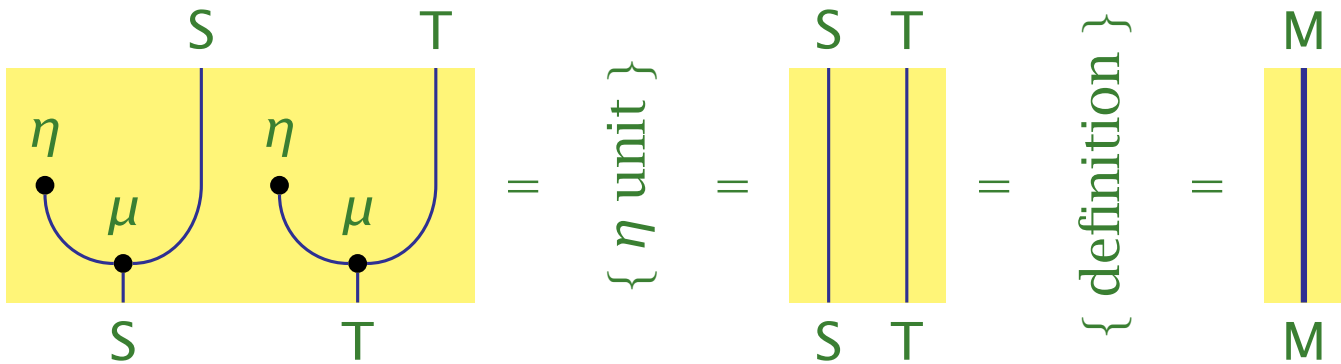
3 Composition of Monads: Unit Law



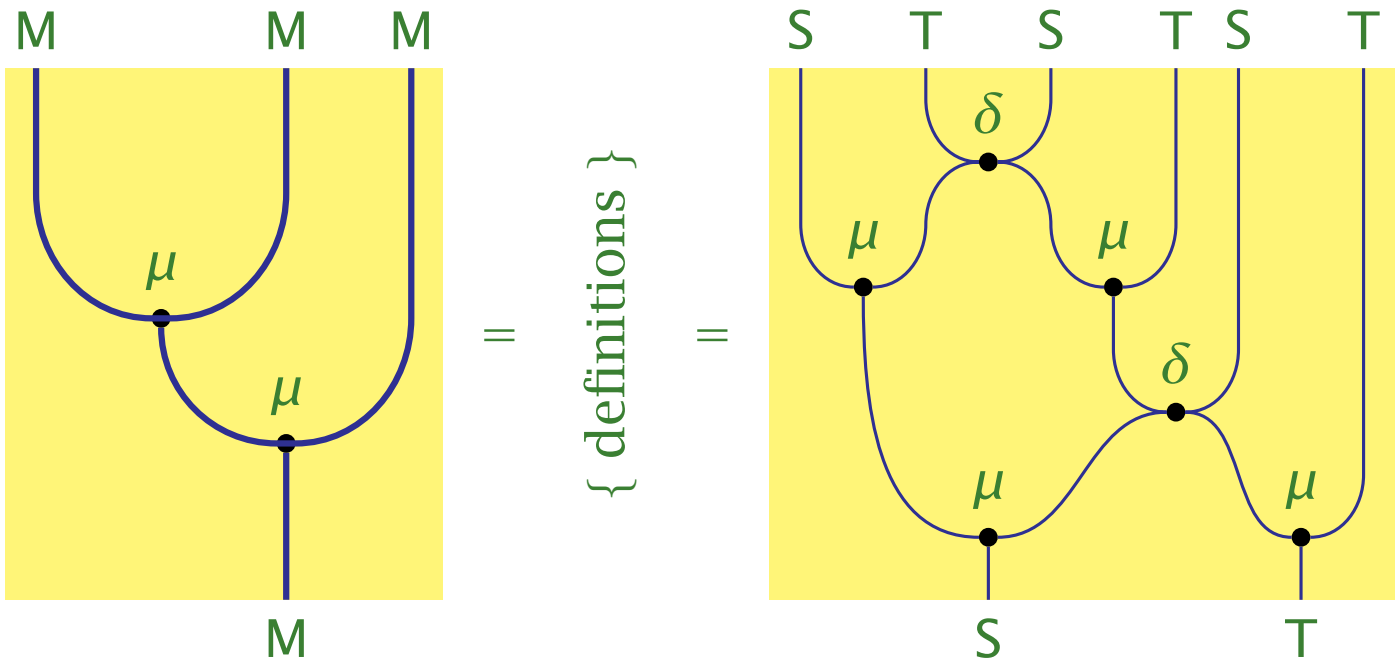
3 Composition of Monads: Unit Law



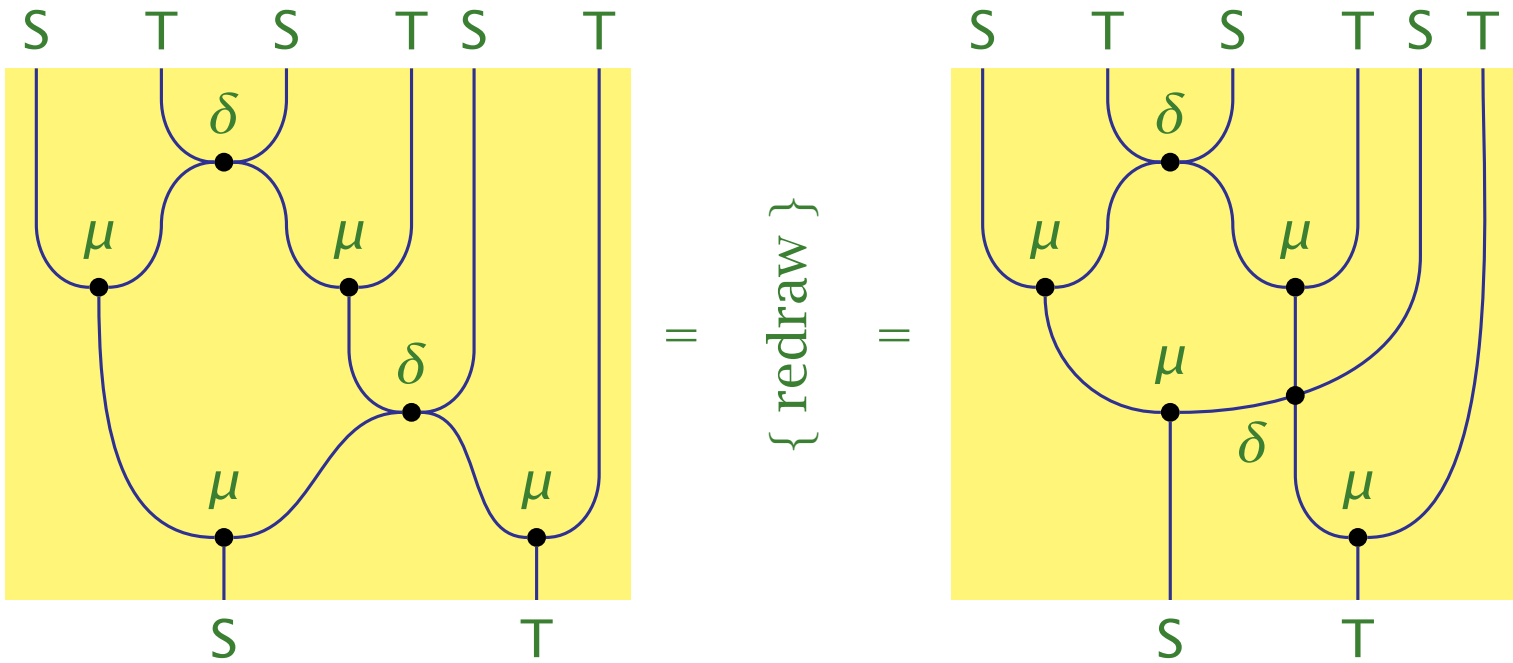
3 Composition of Monads: Unit Law



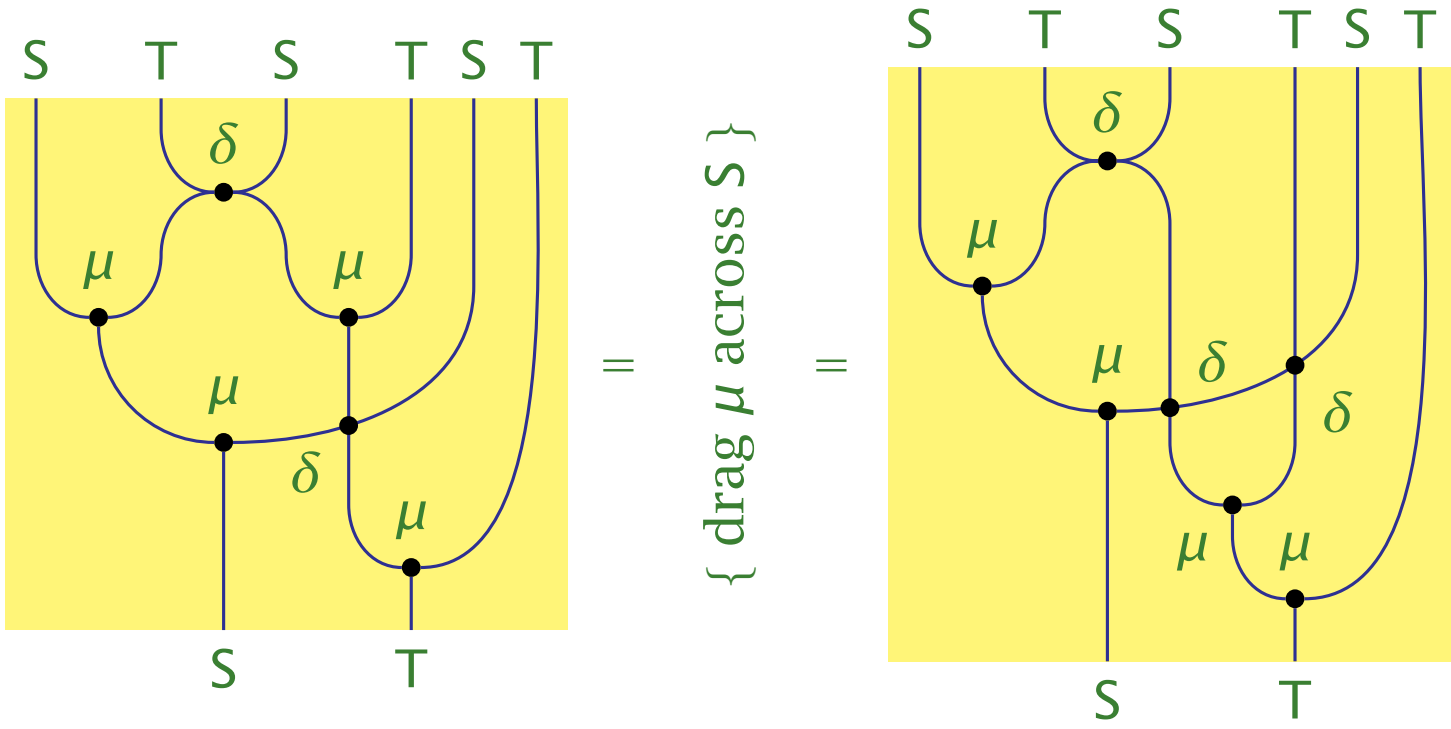
3 Composition of Monads: Associativity



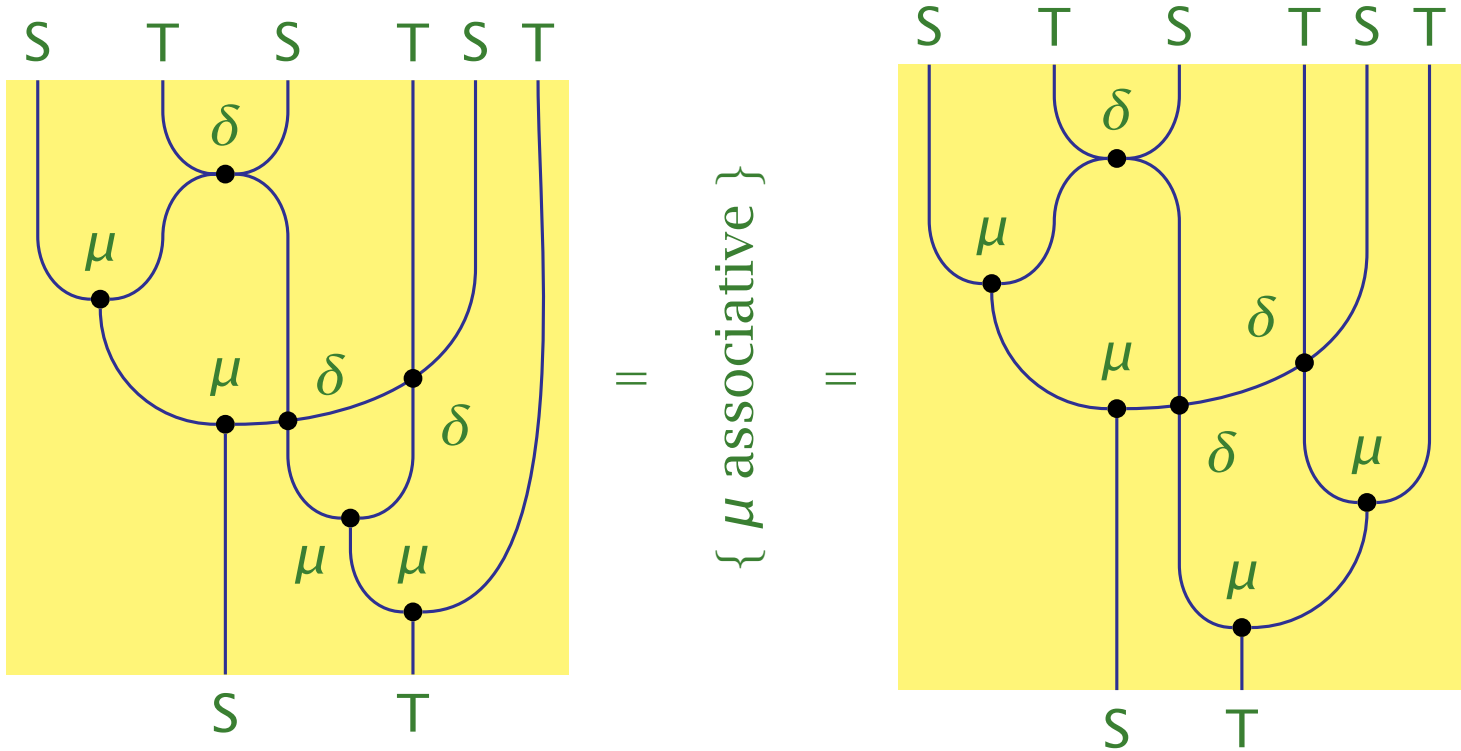
3 Composition of Monads: Associativity



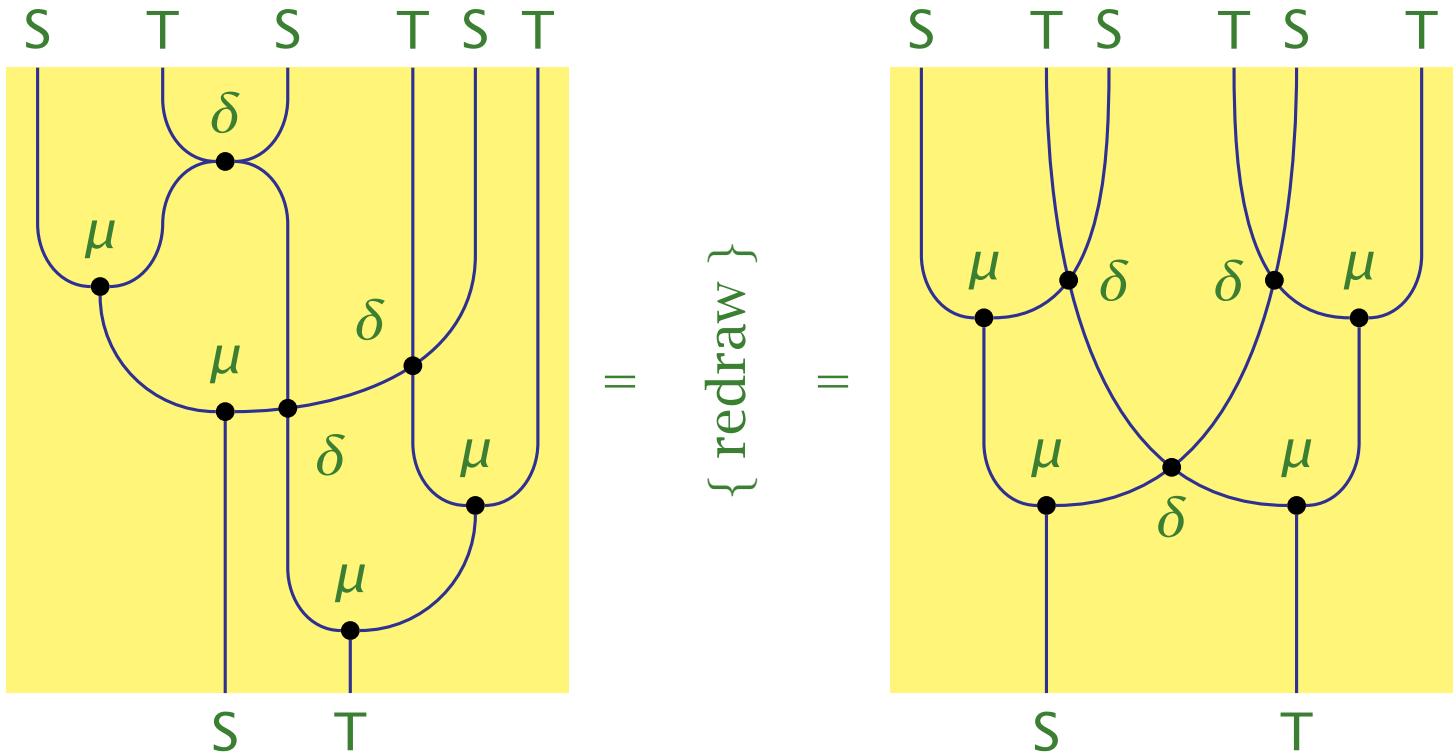
3 Composition of Monads: Associativity



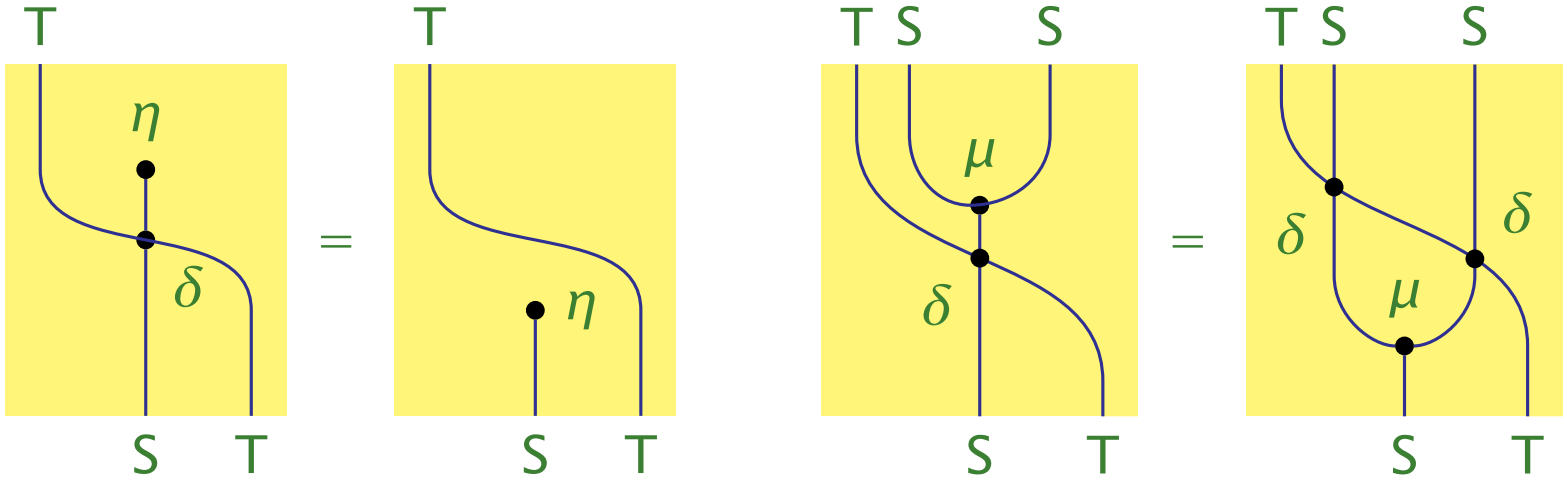
3 Composition of Monads: Associativity

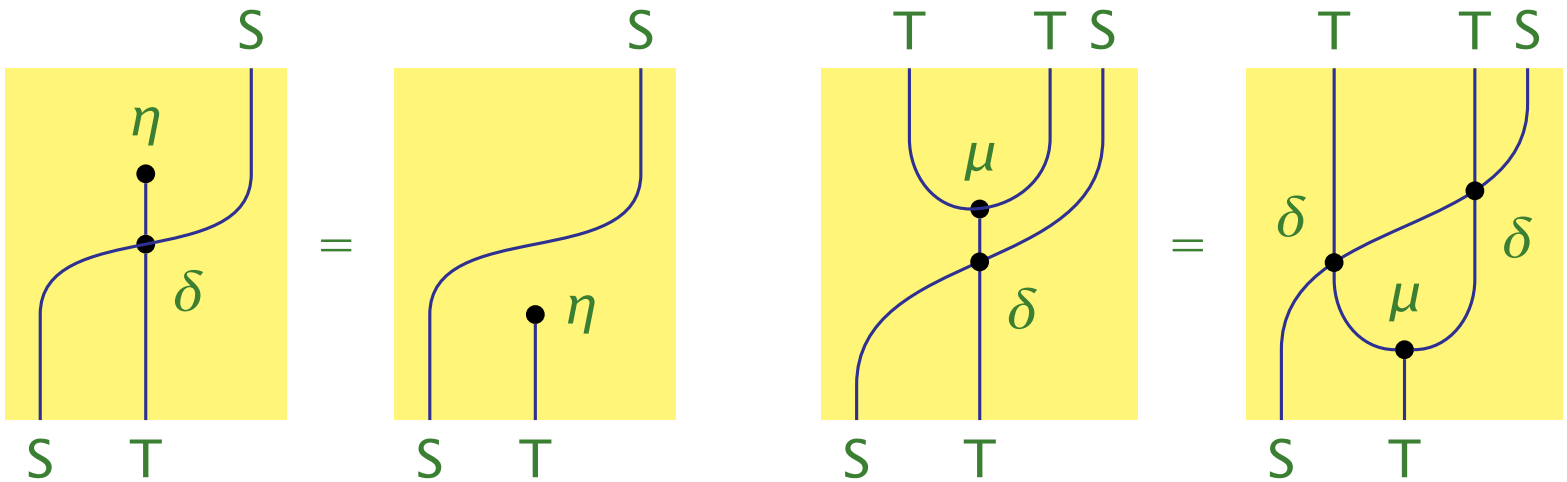


3 Composition of Monads: Associativity



3 Distributive Laws: Dragging Properties



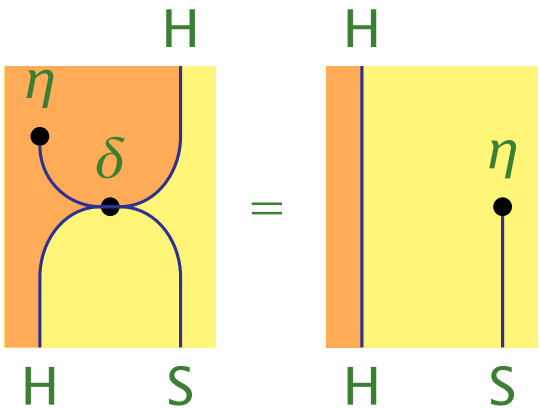


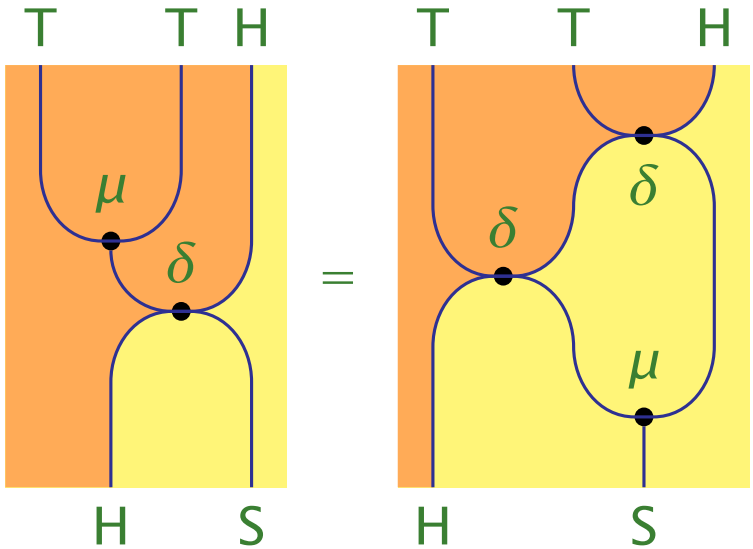
4 EM Laws: Algebraic Definition

$$(\mathsf{S}, \eta, \mu) : \mathcal{C}$$
$$(\mathsf{T}, \eta, \mu) : \mathcal{D}$$
$$\mathsf{H} : \mathcal{C} \to \mathcal{D}$$
$$\delta : \mathsf{T} \circ \mathsf{H} \stackrel{\cdot}{\to} \mathsf{H} \circ \mathsf{S}$$

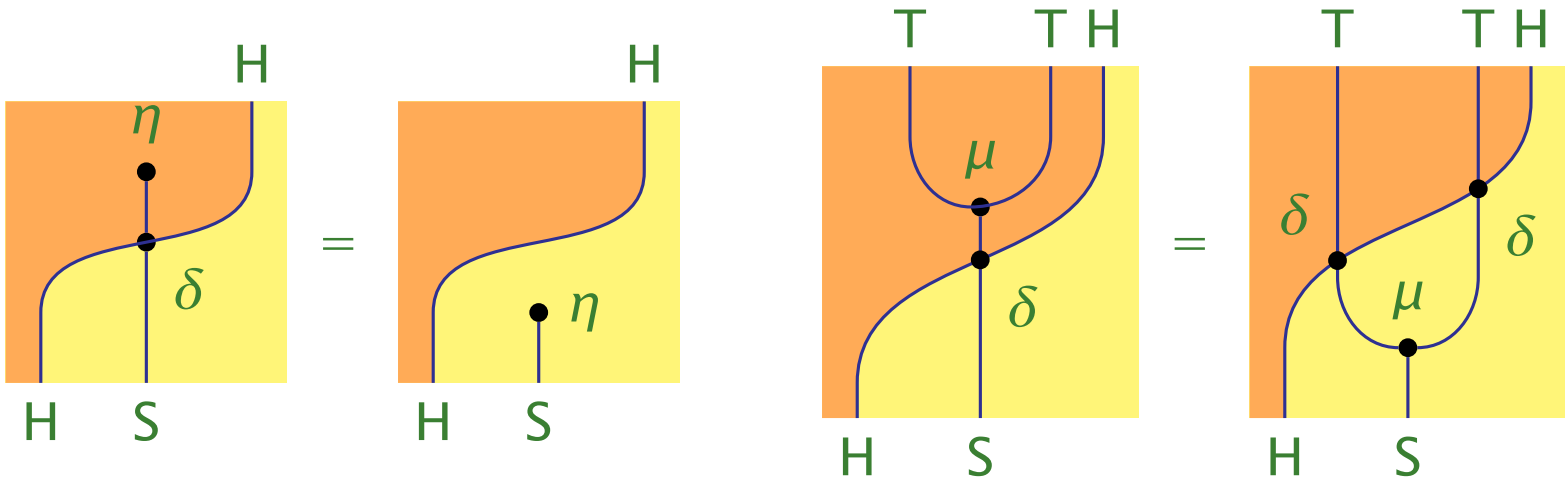
$$\delta \cdot (\eta \circ \mathsf{H}) = (\mathsf{H} \circ \eta)$$
$$\delta \cdot (\mu \circ \mathsf{H}) = (\mathsf{H} \circ \mu) \cdot (\delta \circ \mathsf{S}) \cdot (\mathsf{T} \circ \delta)$$

4 EM Laws: Naive String Diagrams

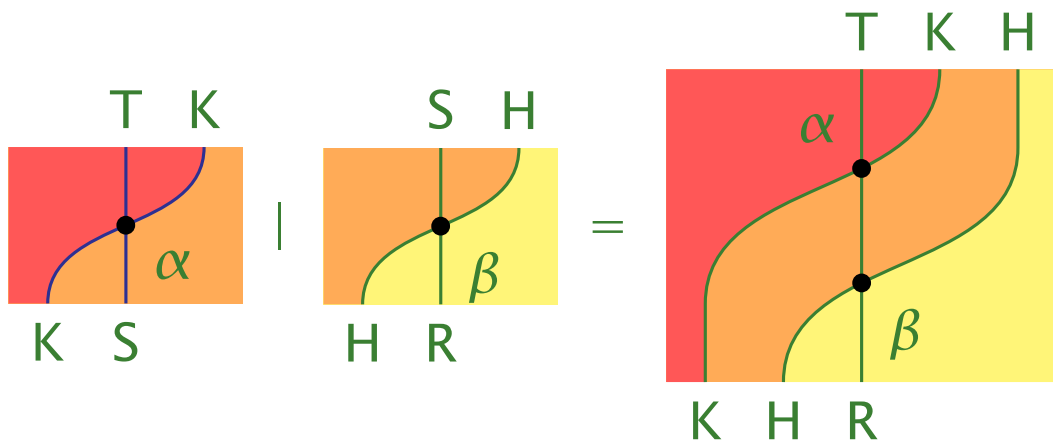




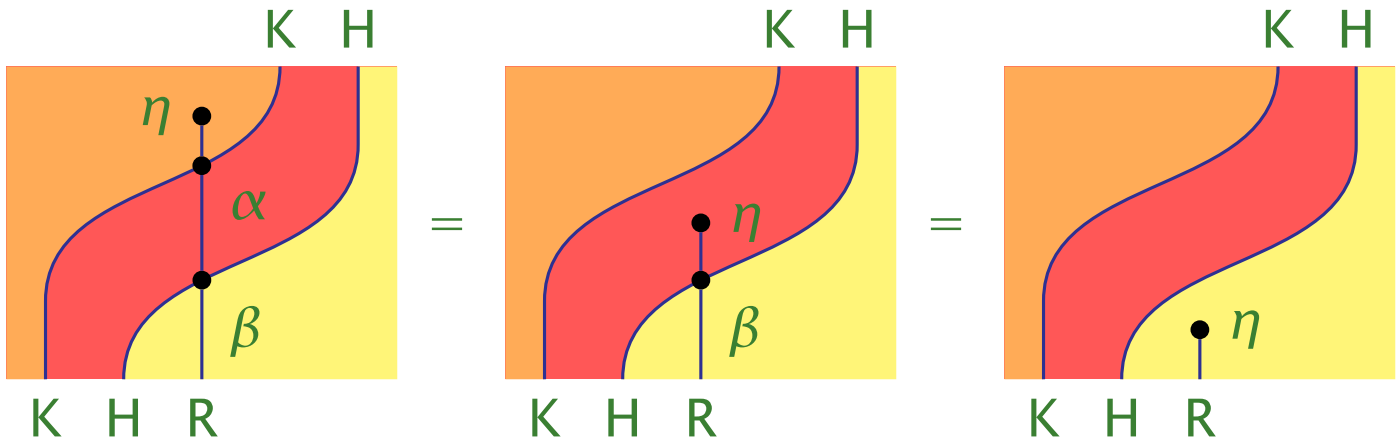
4 EM Laws: String Diagrams



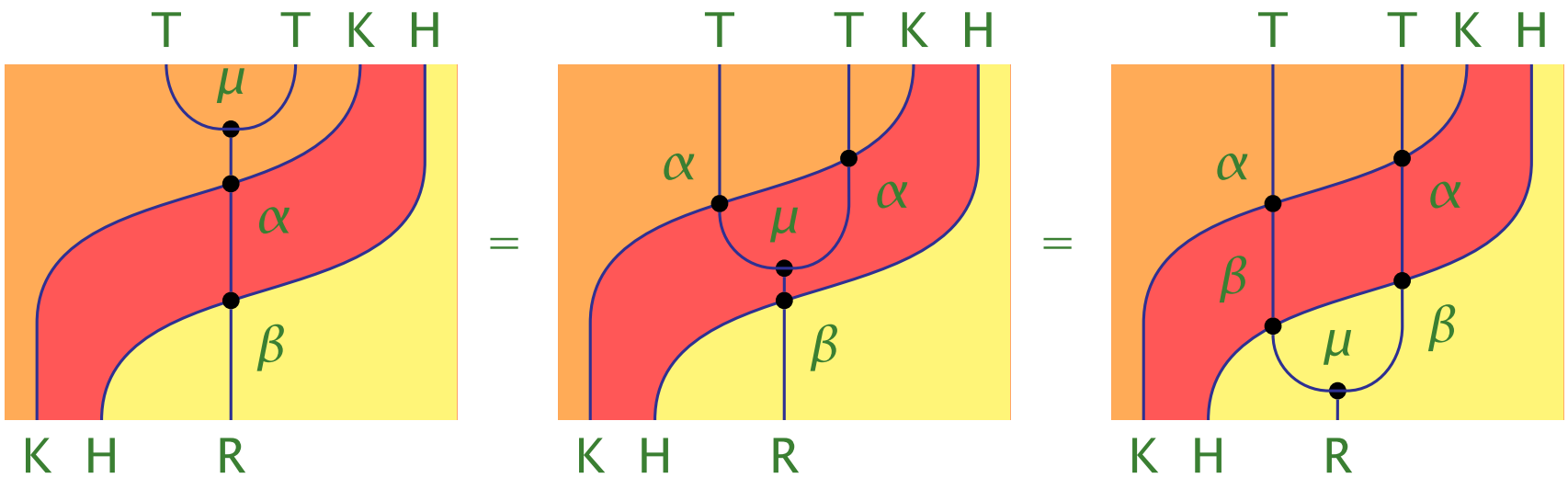
4 EM Laws: Composition



4 EM Laws: Composition



4 EM Laws: Composition



5 Composition of Two Monads

$$\mathsf{T}_1 \circ \mathsf{T}_2$$

$$\delta : \mathsf{T}_2 \circ \mathsf{T}_1 \dot{\to} \mathsf{T}_1 \circ \mathsf{T}_2$$

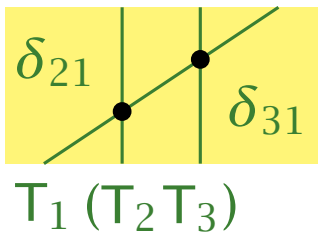
5 Composition of Three Monads

$$T_1 \circ T_2 \circ T_3$$

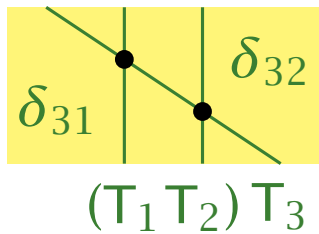
$$\delta_{31} : T_3 \circ T_1 \dot{\to} T_1 \circ T_3 \qquad \delta_{21} : T_2 \circ T_1 \dot{\to} T_1 \circ T_2 \qquad \delta_{32} : T_3 \circ T_2 \dot{\to} T_2 \circ T_3$$

5 Candidates for Distributive Laws

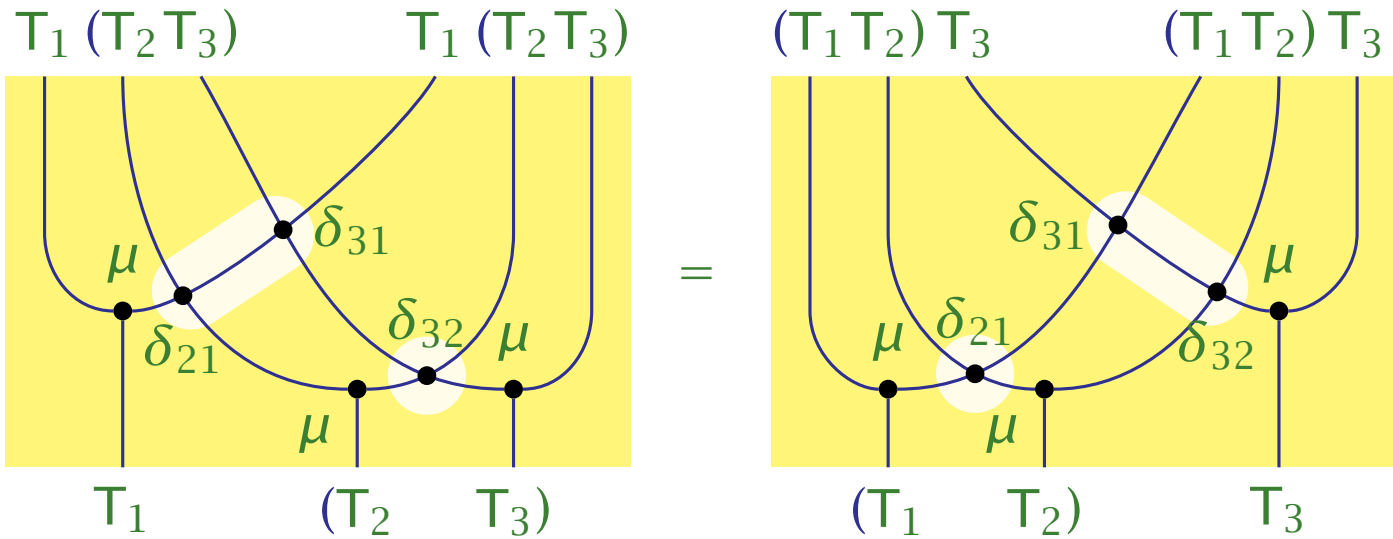
δ =



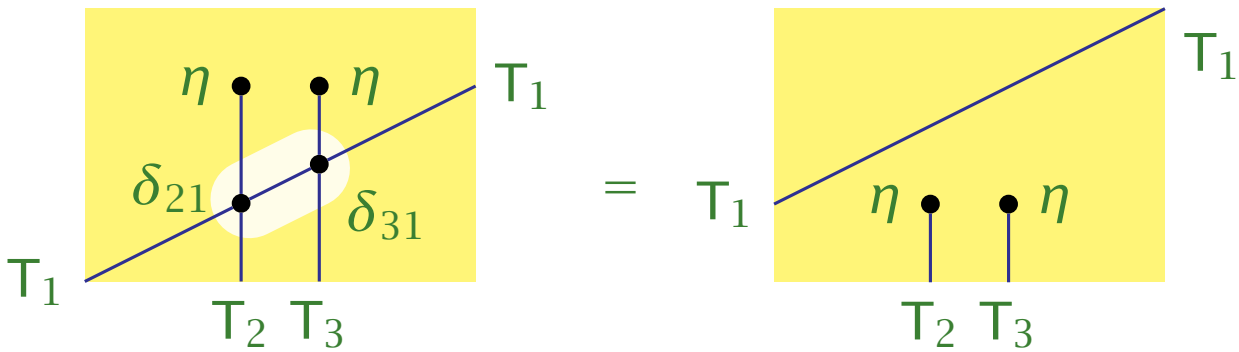
δ' =



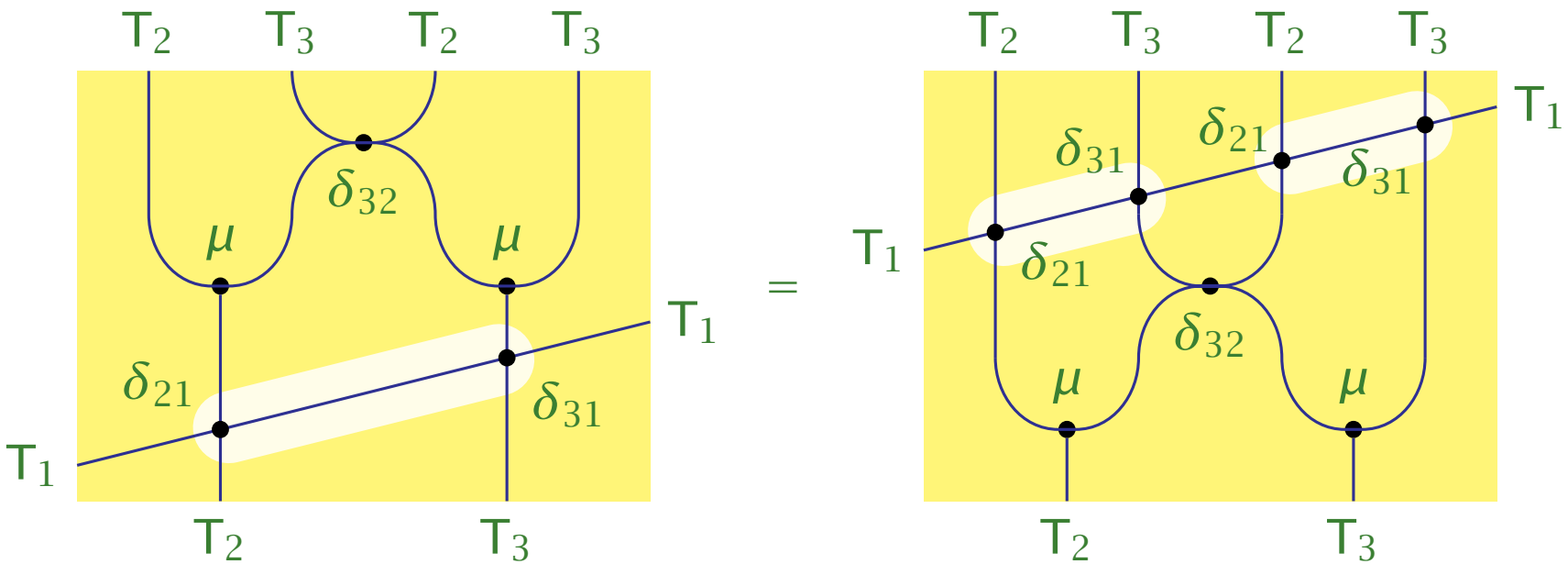
5 Composite Multiplication



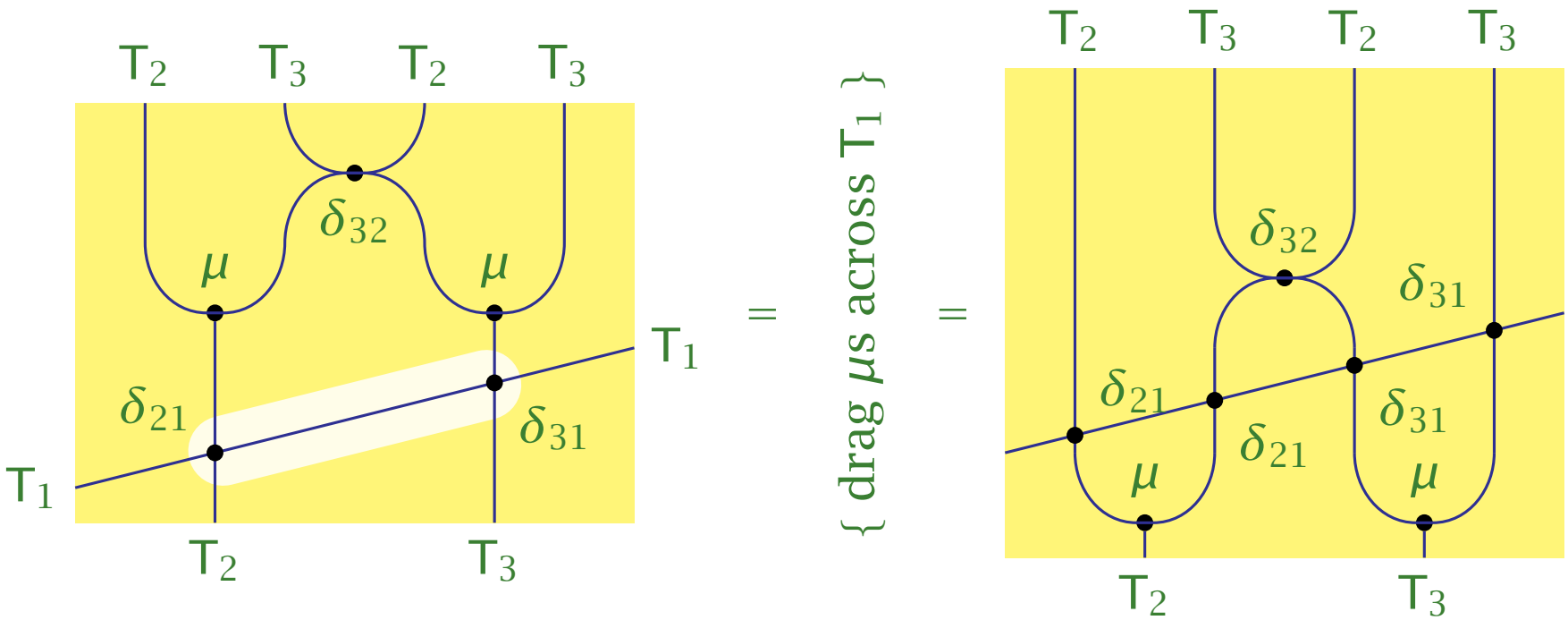
5 Proof Obligation: Dragging the Unit



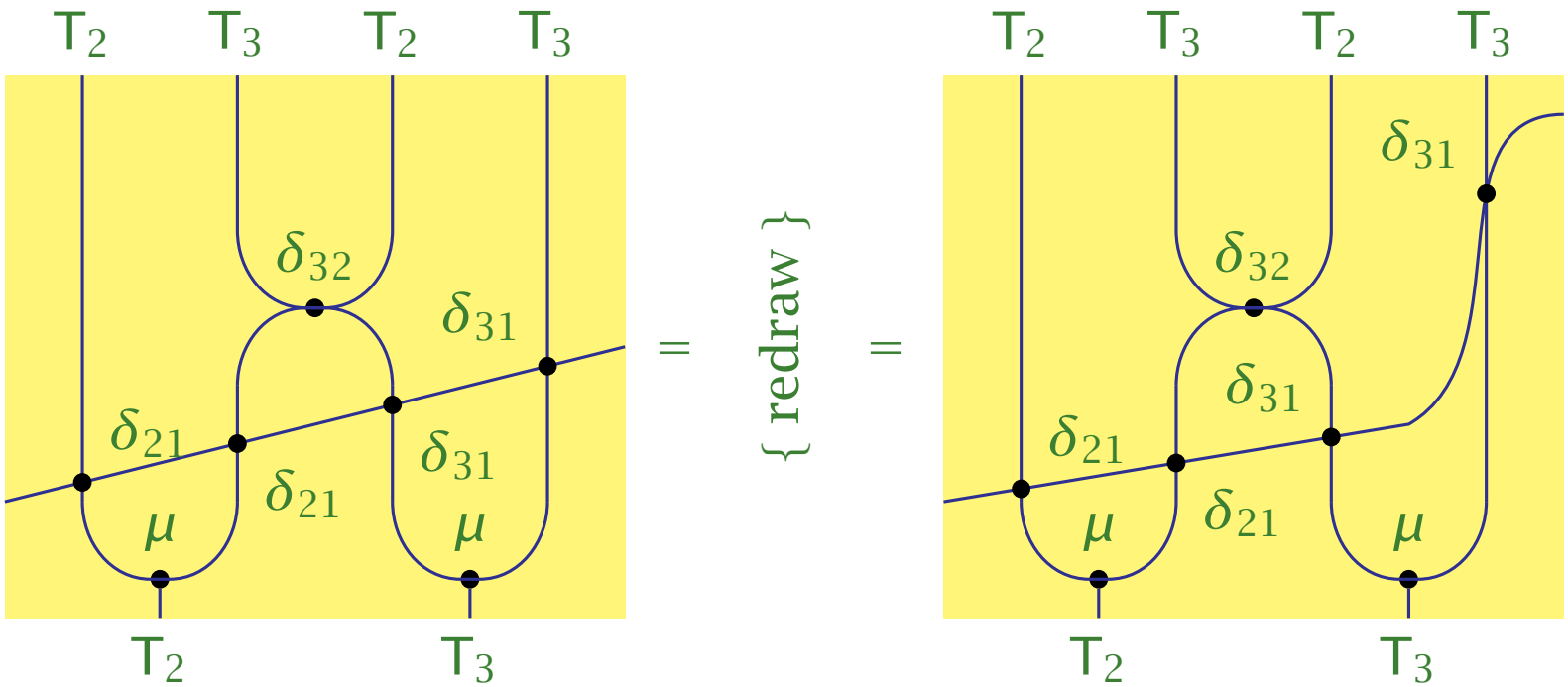
5 Proof Obligation: Dragging the Multiplication



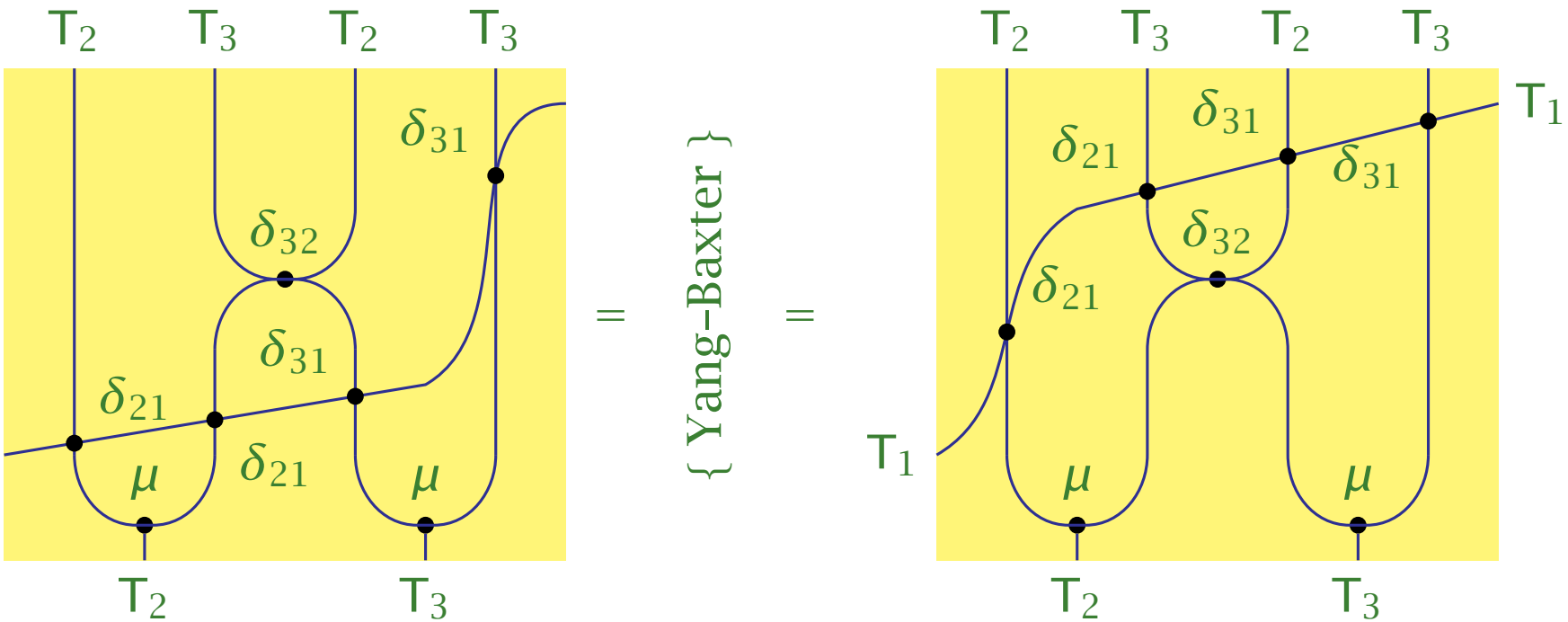
5 Dragging the Multiplication



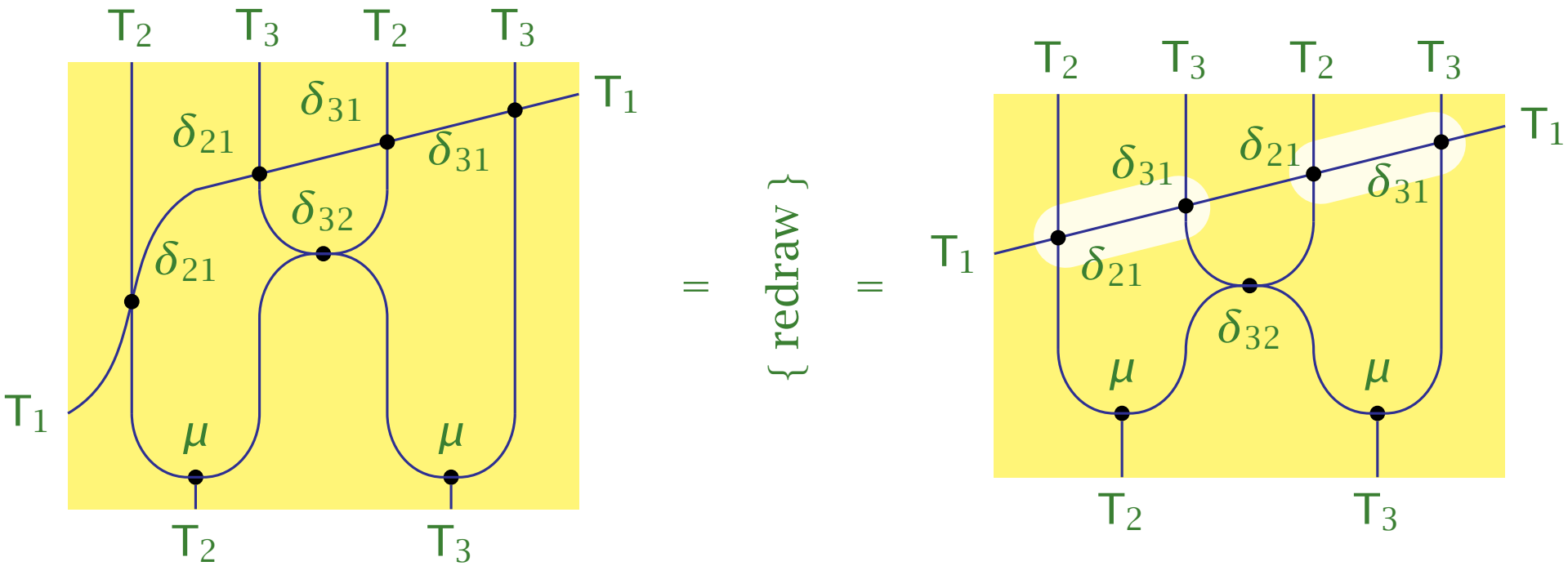
5 Dragging the Multiplication



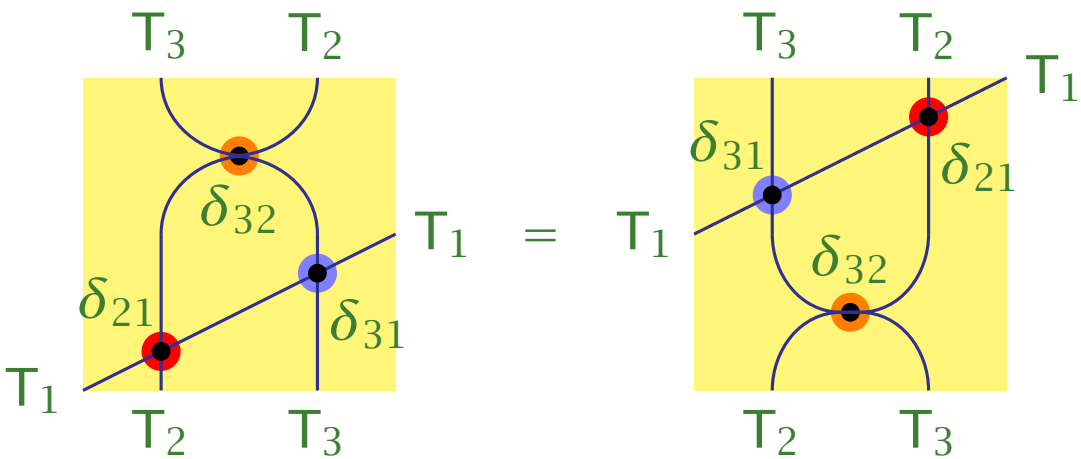
5 Dragging the Multiplication



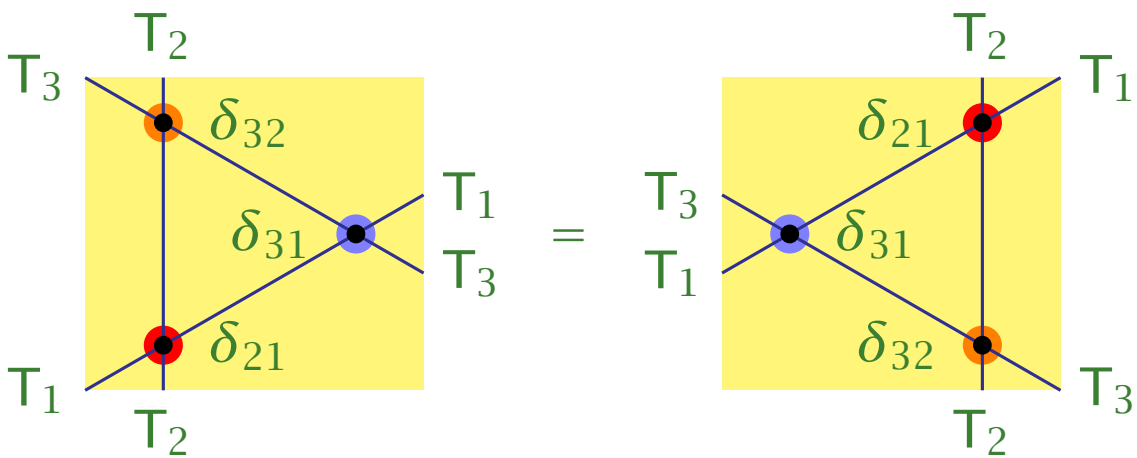
5 Dragging the Multiplication



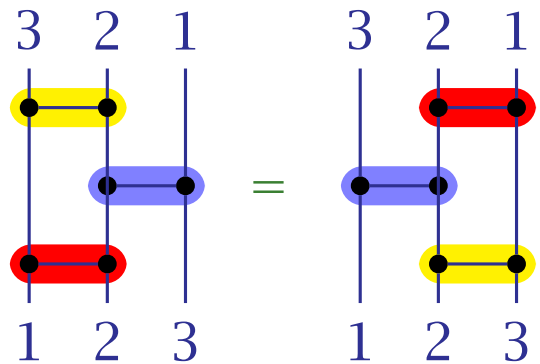
5 Yang-Baxter Equation



5 Yang-Baxter Equation



5 Yang-Baxter Equation and Sorting Networks



6 Conclusion

- String diagrams provide category theory with a new and very distinctive *visual* flavour.
- Drawing string diagrams is an *art*: good diagrammatic choices can make all the difference.
- They turn the stereotype of "definition, theorem, and proof" around, allowing us to *discover* concepts during proof attempts.
- (String diagrams silently deal with distracting bookkeeping steps, such as naturality and functoriality properties.)