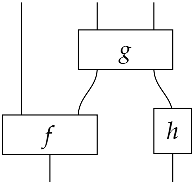
Quantomatic: a proof assistant for diagrammatic reasoning

Aleks Kissinger

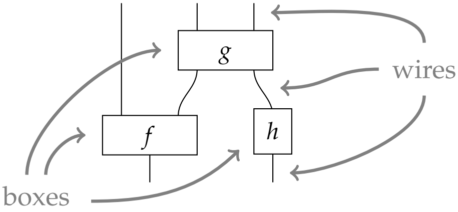
April 11, 2015

- Quantomatic is a *diagrammatic proof assistant*

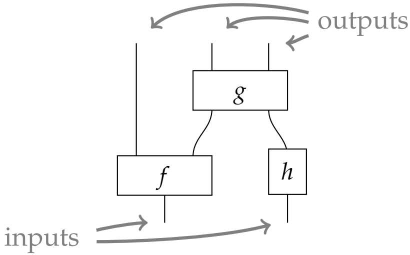
- Quantomatic is a *diagrammatic proof assistant*
- Instead of terms or formulas, its primitive objects are *string diagrams*:



- Quantomatic is a *diagrammatic proof assistant*
- Instead of terms or formulas, its primitive objects are *string diagrams*:

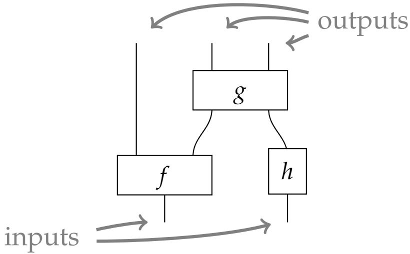


- Quantomatic is a *diagrammatic proof assistant*
- Instead of terms or formulas, its primitive objects are *string diagrams*:



- String diagrams are basically directed (or undirected) graphs, but wires, unlike edges, are allowed to be open, allowing composition (i.e. plugging)

- Quantomatic is a *diagrammatic proof assistant*
- Instead of terms or formulas, its primitive objects are *string diagrams*:



- String diagrams are basically directed (or undirected) graphs, but wires, unlike edges, are allowed to be open, allowing composition (i.e. plugging)
- Proofs are done by substituting sub-diagrams according to *string diagram equations*

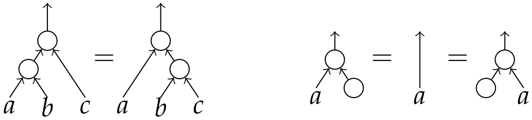
Algebra and rewriting

Algebra and rewriting

- Consider a monoid $(A, \cdot, e)$:

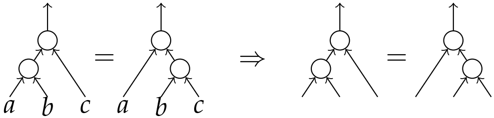
$$(a \cdot b) \cdot c = a \cdot (b \cdot c) \qquad \text{and} \qquad a \cdot e = a = e \cdot a$$

- We could also write these equations using trees:



Algebra and rewriting

- Note we can drop the free variables:



- The role of variables is replaced by the fact that the LHS and RHS have a *shared boundary*:

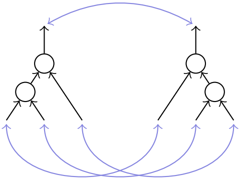


Diagram substitution

- We can apply this rule: '$(a \cdot b) \cdot c = a \cdot (b \cdot c)$' to rewrite a term like this:

$$w \cdot ((x \cdot (y \cdot e)) \cdot z) \quad \longrightarrow \quad w \cdot (x \cdot ((y \cdot e) \cdot z))$$

Diagram substitution

- We can apply this rule: '$(a \cdot b) \cdot c = a \cdot (b \cdot c)$' to rewrite a term like this:

$$w \cdot ((x \cdot (y \cdot e)) \cdot z) \quad \longrightarrow \quad w \cdot (x \cdot ((y \cdot e) \cdot z))$$

- ...or by cutting the LHS directly out of the tree and gluing in the RHS:

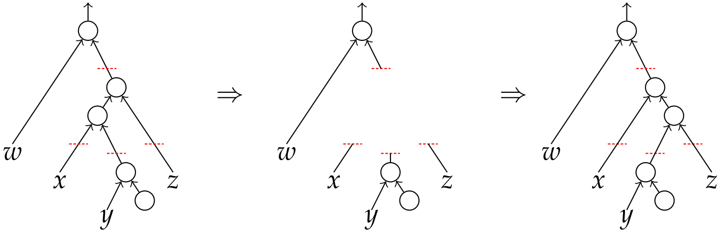
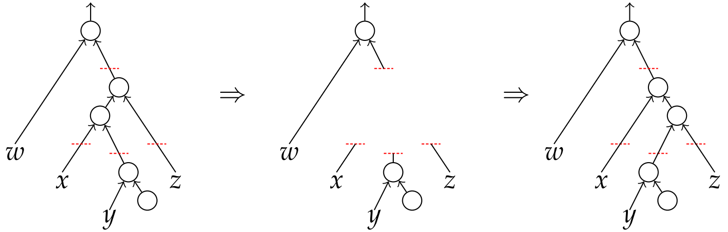


Diagram substitution

- We can apply this rule: '$(a \cdot b) \cdot c = a \cdot (b \cdot c)$' to rewrite a term like this:

$$w \cdot ((x \cdot (y \cdot e)) \cdot z) \quad \longrightarrow \quad w \cdot (x \cdot ((y \cdot e) \cdot z))$$

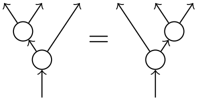
- ...or by cutting the LHS directly out of the tree and gluing in the RHS:

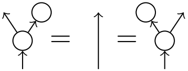


- This treats inputs and outputs symmetrically

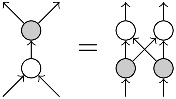
Algebra and coalgebra

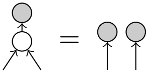
- We can consider structures with many *outputs* as well as inputs.
- *Coalgebraic structures*: algebraic structures "upside-down"

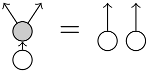




- The most interesting structures consist of algebras *interacting* with coalgebras:

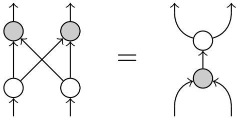




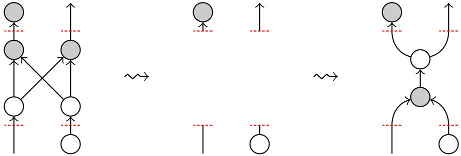


Equational reasoning with diagram substitution

- As before, we can use graphical identities to perform substitutions, but on graphs, rather than trees

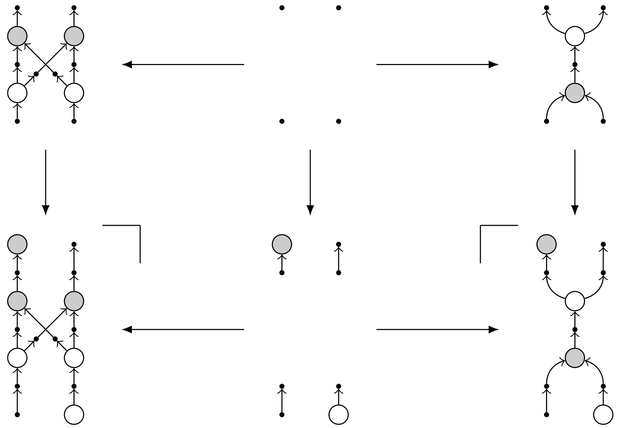


- For example:



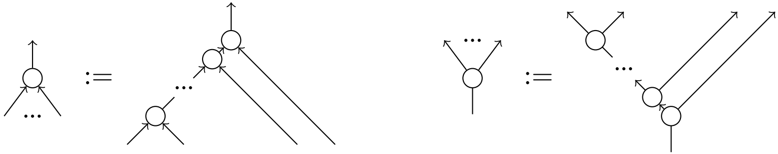
DPO rewriting

- Mechanised by introducing some dummy nodes and doing DPO:

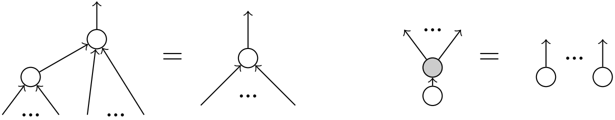


Diagrams with repetition

- If we consider nodes with variable arity, e.g. trees of (co)multiplications:

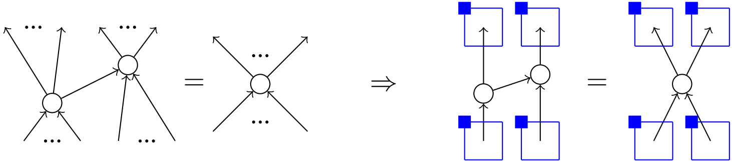


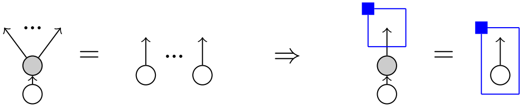
- We can write more general/powerful rules, like:



!-boxes

- We can formalise these equations using !-boxes:

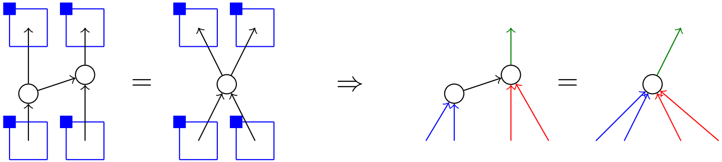




- ...where the box means ‘any number of copies’

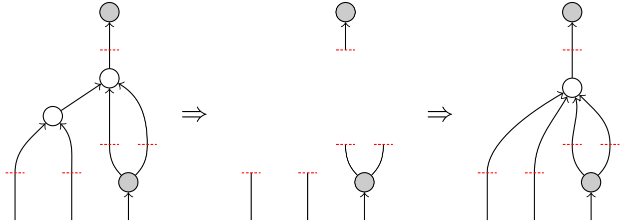
!-box rewriting

- For rewriting, first instantiate:



$$= \qquad \Rightarrow \qquad =$$

- Then apply:



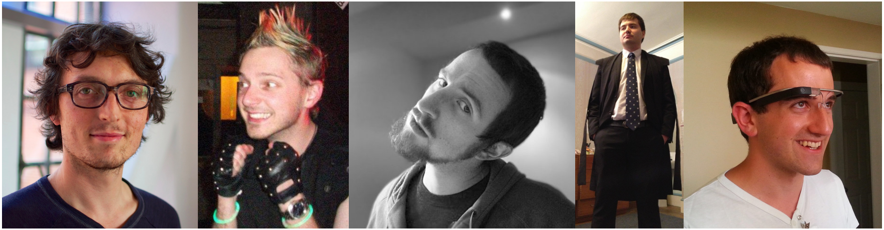
$$\Rightarrow \qquad \Rightarrow$$

Quantomatic demo

- Okay, enough of that...

$$\Longleftarrow$$

Thanks!



- Joint work with Lucas Dixon, Alex Merry, Ross Duncan, Vladimir Zamdzhiev, and David Quick
- See: `quantomatic.github.io`