

Encoding !-Tensors in Quantomatic

David Quick

Oxford Quantum Group

QPL
July 2015



DEPARTMENT OF
**COMPUTER
SCIENCE**



Introduction
●○○○○

!-Tensors
○

!-Graphs (+ordering)
○○

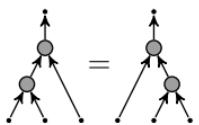
Isomorphism
○○

Summary
○

Diagrammatic Reasoning

Diagrammatic Reasoning

Theory:



Diagrammatic Reasoning

Theory:

$$\begin{array}{c} \text{Diagram 1: } \text{A tree structure with root } A \text{ (top), child } B \text{ (middle), and children } C \text{ and } D \text{ (bottom).} \\ = \\ \text{Diagram 2: } \text{A tree structure with root } A \text{ (top), child } B \text{ (middle), and children } C \text{ and } D \text{ (bottom).} \\ = \\ \text{Diagram 3: } \text{A tree structure with root } A \text{ (top), child } B \text{ (middle), and children } C \text{ and } D \text{ (bottom).} \\ = \\ \text{Diagram 4: } \text{A tree structure with root } A \text{ (top), child } B \text{ (middle), and children } C \text{ and } D \text{ (bottom).} \end{array}$$

Diagrammatic Reasoning

Theory:

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \text{Diagram A} & = & \text{Diagram B} \\ \text{Diagram A} & = & \text{Diagram C} \end{array} \\ \text{Diagram 2: } \begin{array}{c} \text{Diagram D} \\ = \\ \text{Diagram E} \end{array} \end{array}$$

Diagrams A, B, and C are !-Graphs. Diagram A has two nodes with arrows pointing to a top node. Diagram B has two nodes with arrows pointing to a top node, and one additional edge between the two nodes. Diagram C has one node with an arrow pointing to a top node. Diagram D has one node with an arrow pointing to a top node. Diagram E has one node with an arrow pointing to a top node.

Diagrammatic Reasoning

Theory:

$$\begin{array}{c} \text{Diagram 1: } \\ \text{Left: Two nodes, each with two children. The top node has an arrow pointing up.} \\ \text{Right: Two nodes, each with two children. The top node has an arrow pointing up.} \\ = \\ \text{Diagram 2: } \\ \text{Left: A single node with two children. The top child has an arrow pointing up.} \\ \text{Right: A single node with two children. The top child has an arrow pointing up.} \\ = \\ \text{Diagram 3: } \\ \text{Left: A single node with two children. The top child is a circle.} \\ \text{Right: Two nodes, each with one child. The top node has an arrow pointing up.} \\ = \\ \text{Diagram 4: } \\ \text{Left: A single node with two children. The top child is a circle.} \\ \text{Right: Two nodes, each with one child. The top node has an arrow pointing up.} \end{array}$$

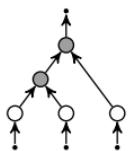
Diagrammatic Reasoning

Theory:

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \text{Tree A} & = & \text{Tree B} \\ \text{Tree A} & = & \text{Tree C} \end{array} \\ \text{Diagram 2: } \begin{array}{ccc} \text{Tree D} & = & \text{Tree E} \\ \text{Tree F} & = & \text{Tree G} \end{array} \end{array}$$

Diagrams 1 and 2 show equational reasoning using trees. Tree A is shown in two different configurations, each equated to another tree. In Diagram 1, Tree A has a central node with three children, one of which is a node with two children. In Diagram 2, Tree D has a central node with three children, one of which is a node with two children. Tree E has a central node with two children, one of which is a leaf node. Tree F has a single leaf node. Tree G has a central node with two children, one of which is a leaf node.

Substitution:



Diagrammatic Reasoning

Theory:

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \text{Tree A} & = & \text{Tree B} \\ \text{Tree A} & = & \text{Tree C} \end{array} \\ \text{Diagram 2: } \begin{array}{ccc} \text{Tree D} & = & \text{Tree E} \\ \text{Tree F} & = & \text{Tree G} \end{array} \end{array}$$

Diagram 1 shows three trees labeled A, B, and C. Tree A has a root node with two children, each with one child. Tree B has a root node with two children, each with one child. Tree C has a root node with one child, which has one child. Diagram 2 shows two pairs of trees labeled D, E, F, and G. Tree D has a root node with one child, which has one child. Tree E has a root node with one child, which has one child. Tree F has a root node with one child, which has one child. Tree G has a root node with one child, which has one child.

Substitution:

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \text{Tree A} & = & \text{Tree B} \\ \text{Tree A} & = & \text{Tree C} \end{array} \\ \text{Diagram 2: } \begin{array}{ccc} \text{Tree D} & = & \text{Tree E} \\ \text{Tree F} & = & \text{Tree G} \end{array} \end{array}$$

Diagram 1 shows three trees labeled A, B, and C. Tree A has a root node with two children, each with one child. Tree B has a root node with two children, each with one child. Tree C has a root node with one child, which has one child. Diagram 2 shows two pairs of trees labeled D, E, F, and G. Tree D has a root node with one child, which has one child. Tree E has a root node with one child, which has one child. Tree F has a root node with one child, which has one child. Tree G has a root node with one child, which has one child.

Diagrammatic Reasoning

Theory:

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \text{Tree A} & = & \text{Tree B} \\ \text{Tree A} & = & \text{Tree C} \end{array} \\ \text{Diagram 2: } \begin{array}{ccc} \text{Tree D} & = & \text{Tree E} \\ \text{Tree F} & = & \text{Tree G} \end{array} \end{array}$$

Diagrammatic reasoning theory examples showing equality between different graph structures.

Substitution:

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \text{Tree A} & = & \text{Tree B} \\ \text{Tree A} & = & \text{Tree C} \end{array} \\ \text{Diagram 2: } \begin{array}{ccc} \text{Tree D} & = & \text{Tree E} \\ \text{Tree F} & = & \text{Tree G} \end{array} \end{array}$$

Diagrammatic reasoning substitution examples showing how one tree structure can be replaced by another.

Diagrammatic Reasoning

Theory:

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \text{Diagram A} & = & \text{Diagram B} \\ \text{Diagram C} & = & \text{Diagram D} \end{array} \\ \text{Diagram 2: } \begin{array}{ccc} \text{Diagram E} & = & \text{Diagram F} \\ \text{Diagram G} & = & \text{Diagram H} \end{array} \end{array}$$

Diagrams A, B, C, D, E, F, G, H are labeled with nodes (circles) and arrows (arrows pointing upwards). Diagram A has two nodes at the top level, each with two children. Diagram B has one node at the top level with three children. Diagram C has one node at the top level with two children. Diagram D has one node at the top level with one child. Diagram E has one node at the top level with two children. Diagram F has one node at the top level with one child. Diagram G has one node at the top level with one child. Diagram H has one node at the top level with two children.

Substitution:

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \text{Diagram A} & = & \text{Diagram B} \\ \text{Diagram C} & = & \text{Diagram D} \end{array} \\ \text{Diagram 2: } \begin{array}{ccc} \text{Diagram E} & = & \text{Diagram F} \end{array} \end{array}$$

Diagrams A, B, C, D, E, F are labeled with nodes (circles) and arrows (arrows pointing upwards). Diagram A has one node at the top level with two children. Diagram B has one node at the top level with one child. Diagram C has one node at the top level with two children. Diagram D has one node at the top level with one child. Diagram E has one node at the top level with two children. Diagram F has one node at the top level with one child.

Commutative:

$$\begin{array}{ccc} \text{Diagram A} & \equiv & \text{Diagram B} \\ \text{Diagram C} & \equiv & \text{Diagram D} \end{array}$$

Diagrams A, B, C, D are labeled with nodes (circles) and arrows (arrows pointing upwards). Diagram A has one node at the top level with two children. Diagram B has one node at the top level with one child. Diagram C has one node at the top level with two children. Diagram D has one node at the top level with one child.

Diagrammatic Reasoning

Commutative

Theory:

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \text{Tree 1} & = & \text{Tree 2} \\ \text{Tree 1} & = & \text{Tree 3} \end{array} \\ \text{Diagram 2: } \begin{array}{ccc} \text{Tree 4} & = & \text{Tree 5} \\ \text{Tree 6} & = & \text{Tree 7} \end{array} \end{array}$$

Substitution:

$$\begin{array}{ccc} \text{Tree 8} & = & \text{Tree 9} \\ \text{Tree 8} & = & \text{Tree 10} \end{array}$$

Commutative:

$$\begin{array}{ccc} \text{Diagram 11} & \equiv & \text{Diagram 12} \\ \text{Diagram 11} & \equiv & \text{Diagram 13} \end{array}$$

Diagrammatic Reasoning

Commutative

Theory:

Diagrams illustrating commutativity in theory:

Row 1: Two trees with shaded gray nodes. The left tree has a root node with two children, each with one child. The right tree has a root node with two children, each with one child.

Row 2: Two trees with white circle nodes. The left tree has a root node with two children, each with one child. The right tree has a root node with two children, each with one child.

Non-Commutative

Substitution:

Diagrams illustrating substitution in non-commutative reasoning:

Three trees with a central shaded gray node and three white circle children. The first tree has solid lines connecting the center to the children. The second tree has dashed lines connecting the center to the children. The third tree has solid lines connecting the center to the children.

Commutative:

Diagrams illustrating commutativity in non-commutative reasoning:

Three trees with a central shaded gray node and three white circle children. The first tree has solid lines connecting the center to the children. The second tree has dashed lines connecting the center to the children. The third tree has solid lines connecting the center to the children.

Diagrammatic Reasoning

Commutative

Theory:

Two rows of diagrams illustrating commutativity. The top row shows two trees where nodes are swapped horizontally. The bottom row shows two trees where nodes are swapped vertically.

Non-Commutative

Substitution:

Three trees illustrating substitution. The first tree has a central node with three children. The second tree has a central node with three children, where one child is replaced by a different node. The third tree has a central node with three children, where another child is replaced by a different node.

Commutative:

Three trees illustrating commutativity in substitution. The first tree has a central node with three children. The second tree has a central node with three children, where one child is replaced by a different node. The third tree has a central node with three children, where another child is replaced by a different node.

Non-commutative:

Three trees illustrating non-commutativity in substitution. The first tree has a central node with three children. The second tree has a central node with three children, where one child is replaced by a different node. The third tree has a central node with three children, where another child is replaced by a different node.

Diagrammatic Reasoning

Commutative

Theory:

Two rows of diagrams illustrating commutativity. The top row shows two trees where the nodes are swapped horizontally. The bottom row shows two trees where the nodes are swapped vertically.

Non-Commutative

Substitution:

Three diagrams illustrating substitution in a non-commutative setting. Each diagram shows a tree with a shaded node being replaced by an unshaded node.

Commutative:

Three diagrams illustrating commutativity in substitution. Each diagram shows a tree with a shaded node being replaced by an unshaded node, demonstrating that the order of substitution does not matter.

Non-commutative:

Three diagrams illustrating non-commutativity in substitution. Each diagram shows a tree with a shaded node being replaced by an unshaded node, demonstrating that the order of substitution matters.

Diagrammatic Reasoning

Commutative

Theory:

Two rows of diagrams illustrating commutative properties. The top row shows two sets of diagrams where the order of operations or components does not matter. The bottom row shows two sets of diagrams where the order of components matters.

Top row (Tensor-like):
Left: A diagram with a central node connected to three nodes below it, with arrows pointing upwards. Right: A diagram with a central node connected to three nodes below it, with arrows pointing upwards.
Left: A diagram with a central node connected to three nodes below it, with arrows pointing upwards. Right: A diagram with a central node connected to three nodes below it, with arrows pointing upwards.

Bottom row (Graph-like):
Left: A diagram with a central node connected to three nodes below it, with arrows pointing upwards. Right: A diagram with a central node connected to three nodes below it, with arrows pointing upwards.
Left: A diagram with a central node connected to three nodes below it, with arrows pointing upwards. Right: A diagram with a central node connected to three nodes below it, with arrows pointing upwards.

Substitution:

Three diagrams showing the substitution of one tensor/generator for another in a commutative context. The first diagram shows a central node with three outgoing arrows being replaced by a node with three outgoing arrows. The second diagram shows a central node with three outgoing arrows being replaced by a node with three outgoing arrows. The third diagram shows a central node with three outgoing arrows being replaced by a node with three outgoing arrows.

Commutative:

Three diagrams illustrating commutative properties of graphs with self-loops. The first diagram shows a central node with two outgoing arrows and one self-loop arrow. The second diagram shows a central node with two outgoing arrows and one self-loop arrow. The third diagram shows a central node with two outgoing arrows and one self-loop arrow.

Non-Commutative

Theory:

Two diagrams illustrating non-commutative properties. The left diagram shows a central node with three outgoing arrows, with arrows pointing upwards. The right diagram shows a central node with three outgoing arrows, with arrows pointing upwards.

Non-commutative:

Three diagrams illustrating non-commutative properties of graphs with self-loops. The first diagram shows a central node with two outgoing arrows and one self-loop arrow. The second diagram shows a central node with two outgoing arrows and one self-loop arrow. The third diagram shows a central node with two outgoing arrows and one self-loop arrow.

Diagrammatic Reasoning

Commutative

Theory:

Two rows of diagrams illustrating commutative properties. The top row shows two sets of diagrams where the order of operations or connections does not matter. The bottom row shows two sets of diagrams where the order of elements (circles) matters.

Substitution:

Three diagrams showing the substitution of one element for another in a commutative structure, resulting in equivalent configurations.

Commutative:

Three diagrams illustrating commutative properties, showing that different orders of operations or connections result in equivalent configurations.

Non-Commutative

Theory:

Two rows of diagrams illustrating non-commutative properties. The top row shows two sets of diagrams where the order of operations or connections matters. The bottom row shows two sets of diagrams where the order of elements (circles) matters.

Non-commutative:

Three diagrams illustrating non-commutative properties, showing that different orders of operations or connections result in non-equivalent configurations.

Diagrammatic Reasoning

Commutative

Theory:

Two rows of diagrams illustrating commutative properties. The top row shows two sets of diagrams where the order of operations or components does not matter. The bottom row shows two sets of diagrams where the order of components matters.

Substitution:

Three diagrams showing the substitution of one tensor or graph structure for another within a larger structure, demonstrating that the overall commutative property is preserved under such substitutions.

Commutative:

Three diagrams illustrating commutative properties, showing that different orders of components or operations result in equivalent structures.

Non-Commutative

Theory:

Two rows of diagrams illustrating non-commutative properties. The top row shows two sets of diagrams where the order of operations or components matters, and the equality signs are replaced by ≠ symbols. The bottom row shows two sets of diagrams where the order of components matters.

Non-commutative:

Three diagrams illustrating non-commutative properties, showing that different orders of components or operations result in inequivalent structures.

Diagrammatic Reasoning

Commutative

Theory:

Two rows of diagrams illustrating commutative properties. The top row shows two sets of diagrams where the order of operations (arrows) does not matter. The bottom row shows two sets of diagrams where the order of elements (circles) does not matter.

Non-Commutative

Theory:

Two rows of diagrams illustrating non-commutative properties. The top row shows two sets of diagrams where the order of operations (arrows) matters. The bottom row shows two sets of diagrams where the order of elements (circles) matters.

Substitution:

Three diagrams showing the equivalence of different configurations of nodes and edges, demonstrating how one can be substituted for another while maintaining equality.

Commutative:

Three diagrams showing configurations of nodes and edges that are equivalent due to commutativity, with arrows indicating the direction of equivalence.

Non-commutative:

Three diagrams showing configurations of nodes and edges that are not equivalent due to non-commutativity, with arrows indicating the direction of inequality.

Diagrammatic Reasoning

Commutative

Theory:

Two rows of diagrams illustrating commutative properties. The top row shows two sets of diagrams where the order of operations or components does not matter. The bottom row shows two sets of diagrams where the order of components matters, resulting in different outcomes.

Substitution:

Three diagrams showing how a tensor or graph structure can be substituted for another, maintaining the overall commutative property.

Commutative:

Three diagrams showing commutative properties of specific tensor configurations involving loops and connections.

Non-Commutative

Theory:

Two rows of diagrams illustrating non-commutative properties. The top row shows two sets of diagrams where the order of operations or components matters, resulting in different outcomes. The bottom row shows two sets of diagrams where the order of components matters, resulting in different outcomes.

Substitution:

A single diagram showing a substitution rule for a non-commutative structure, where the overall non-commutative nature is preserved.

Non-commutative:

Three diagrams showing non-commutative properties of specific tensor configurations involving loops and connections, where the order of components is crucial.

Diagrammatic Reasoning

Commutative

Theory:

Two rows of diagrams illustrating commutative properties. The top row shows two sets of diagrams where the order of operations or connections does not matter. The bottom row shows two sets of diagrams where the order of elements (circles) matters.

Substitution:

Three diagrams showing the equivalence of different tensor or graph configurations through substitution.

Commutative:

Three diagrams showing the equivalence of different tensor or graph configurations through commutative properties.

Non-Commutative

Theory:

Two rows of diagrams illustrating non-commutative properties. The top row shows two sets of diagrams where the order of operations or connections matters. The bottom row shows two sets of diagrams where the order of elements (circles) matters.

Substitution:

Two diagrams showing the equivalence of different non-commutative tensor or graph configurations through substitution.

Non-commutative:

Three diagrams showing the non-equivalence of different non-commutative tensor or graph configurations.

Diagrammatic Reasoning

Commutative

Theory:

Two rows of diagrams illustrating commutative properties. The top row shows two sets of diagrams where the order of operations or connections does not matter. The bottom row shows two sets of diagrams where the order of elements (circles) matters.

Substitution:

Three diagrams showing how a tensor or graph structure can be substituted for another. The first diagram shows a tensor with two inputs and one output being replaced by a tensor with three inputs and one output. The second diagram shows a tensor with three inputs and one output being replaced by a tensor with two inputs and one output. The third diagram shows a tensor with two inputs and one output being replaced by a tensor with two inputs and one output.

Commutative:

Three diagrams illustrating commutative properties. The first diagram shows a tensor with two inputs and one output. The second diagram shows a tensor with two inputs and one output. The third diagram shows a tensor with two inputs and one output.

Non-Commutative

Theory:

Two rows of diagrams illustrating non-commutative properties. The top row shows two sets of diagrams where the order of operations or connections matters. The bottom row shows two sets of diagrams where the order of elements (circles) matters.

Substitution:

Three diagrams showing how a tensor or graph structure can be substituted for another. The first diagram shows a tensor with two inputs and one output being replaced by a tensor with three inputs and one output. The second diagram shows a tensor with three inputs and one output being replaced by a tensor with two inputs and one output. The third diagram shows a tensor with two inputs and one output being replaced by a tensor with two inputs and one output.

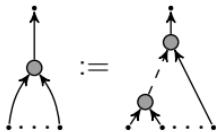
Non-commutative:

Three diagrams illustrating non-commutative properties. The first diagram shows a tensor with two inputs and one output. The second diagram shows a tensor with two inputs and one output. The third diagram shows a tensor with two inputs and one output.

Families of (Commutative) Diagrams

Families of (Commutative) Diagrams

Arbitrary arity:



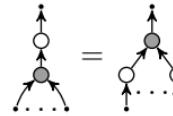
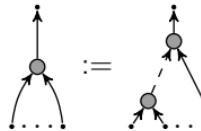
Families of (Commutative) Diagrams

Arbitrary arity:



Families of (Commutative) Diagrams

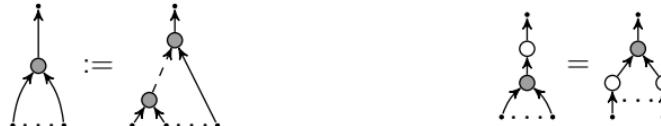
Arbitrary arity:



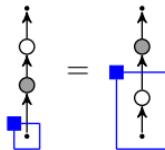
!-Boxes

Families of (Commutative) Diagrams

Arbitrary arity:



!-Boxes

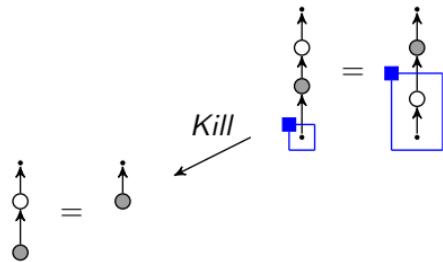


Families of (Commutative) Diagrams

Arbitrary arity:



!-Boxes

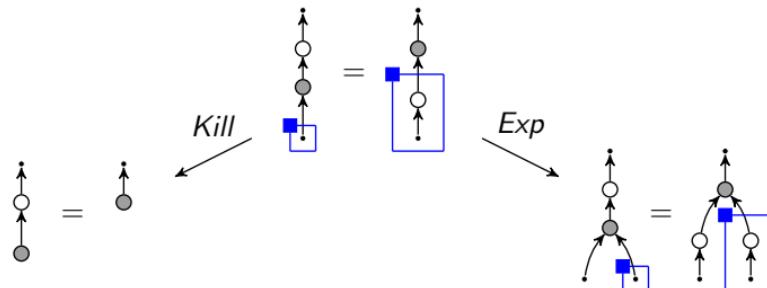


Families of (Commutative) Diagrams

Arbitrary arity:



!-Boxes

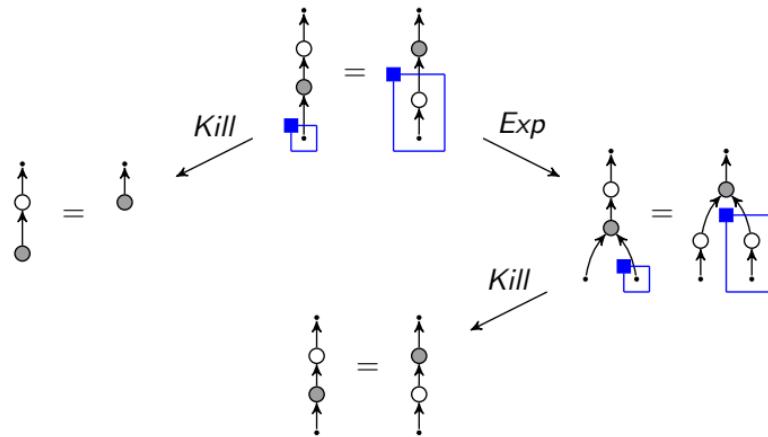


Families of (Commutative) Diagrams

Arbitrary arity:



!-Boxes

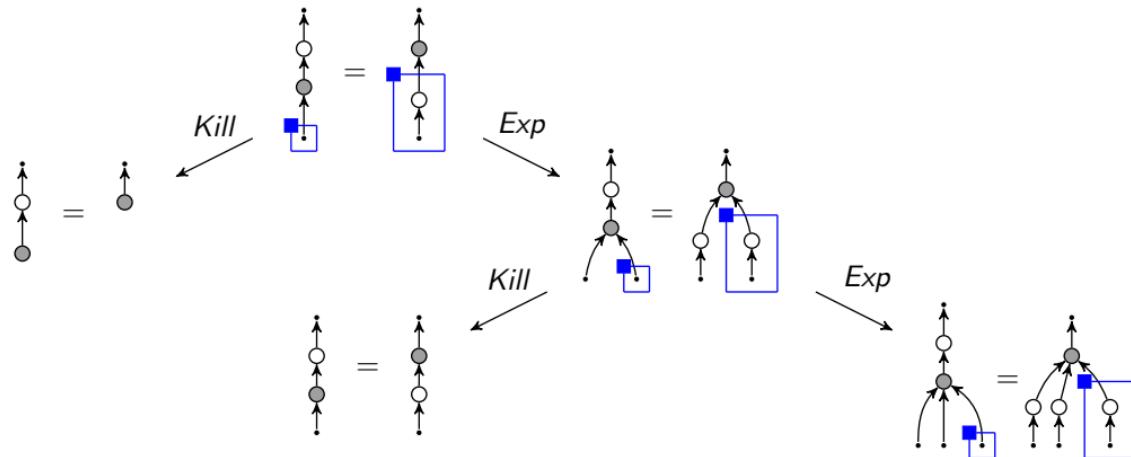


Families of (Commutative) Diagrams

Arbitrary arity:



!-Boxes

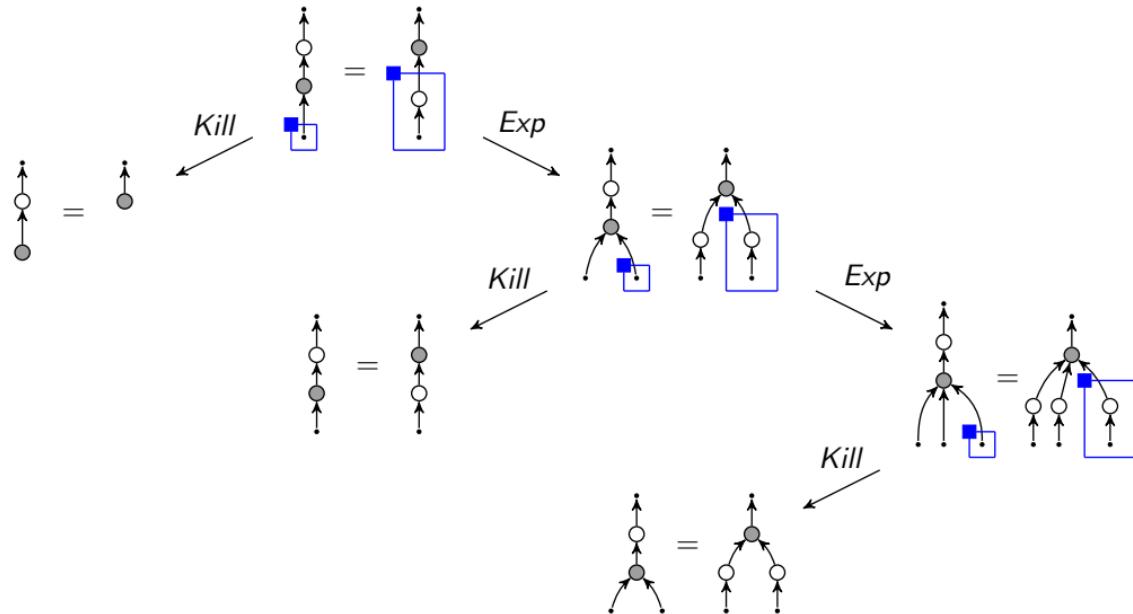


Families of (Commutative) Diagrams

Arbitrary arity:



!-Boxes

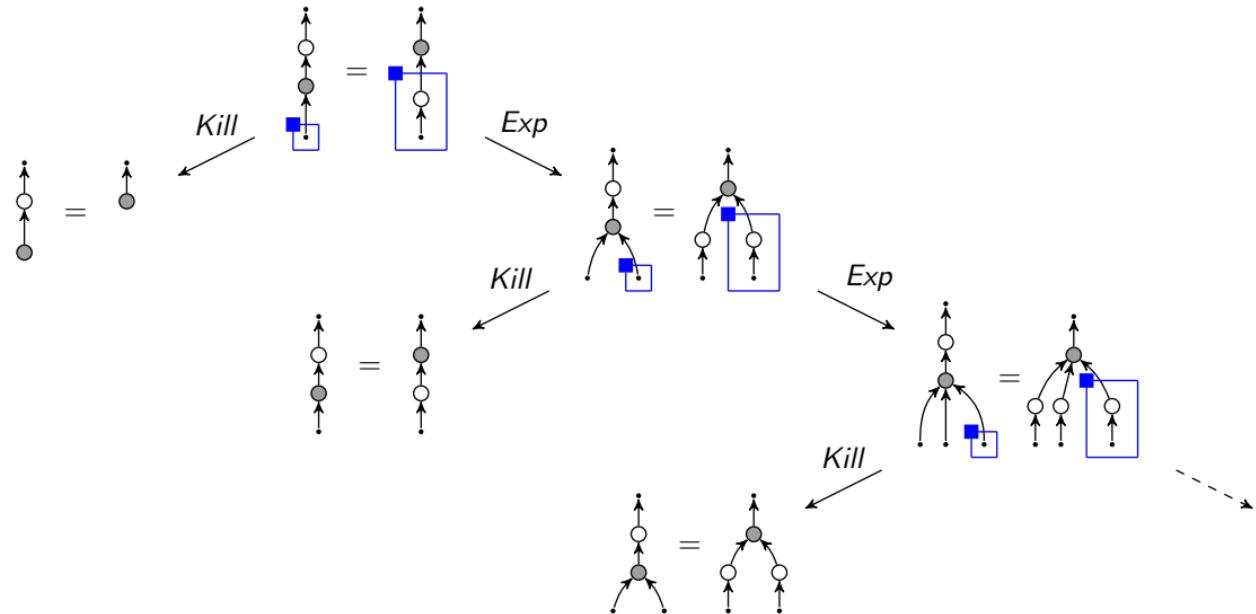


Families of (Commutative) Diagrams

Arbitrary arity:



!-Boxes

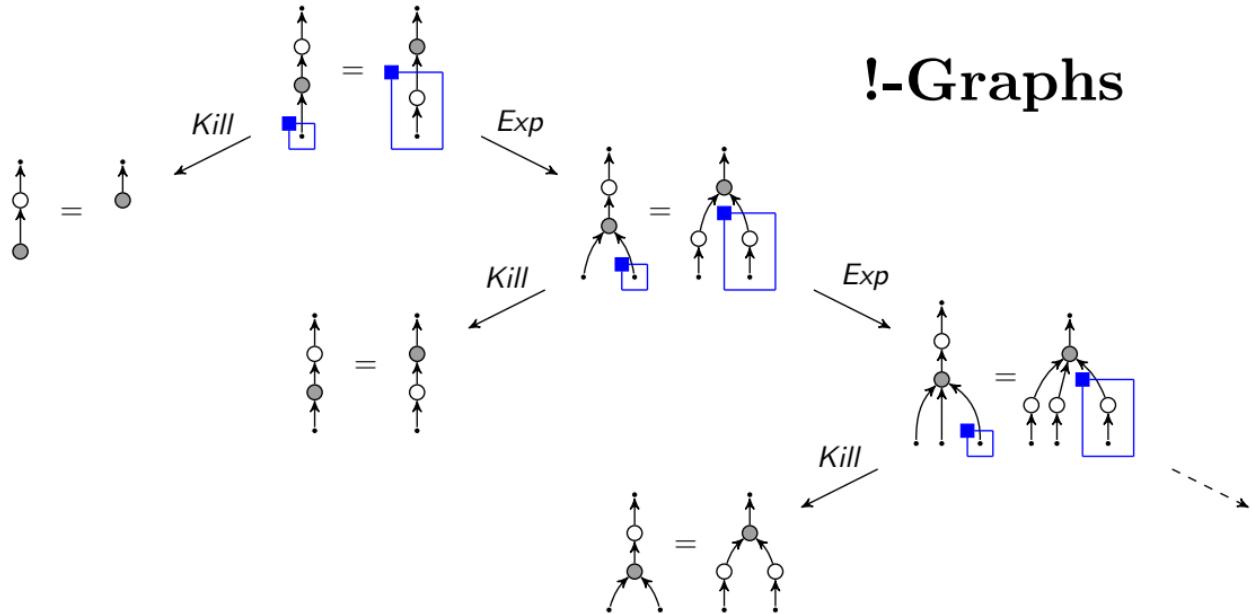


Families of (Commutative) Diagrams

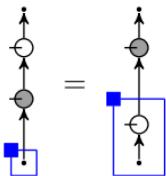
Arbitrary arity:



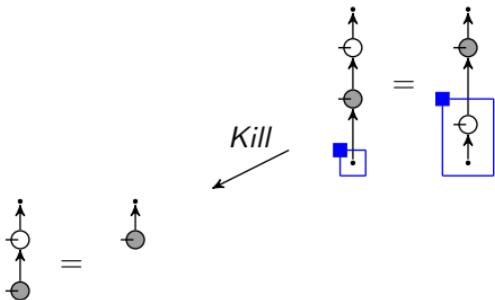
!-Boxes



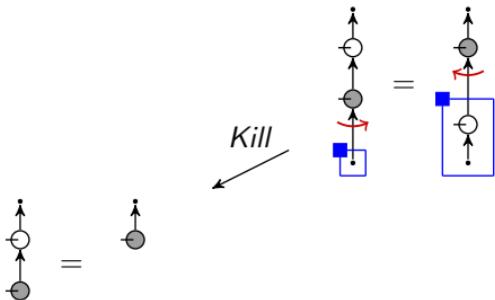
Families of (Non-Commutative) Diagrams



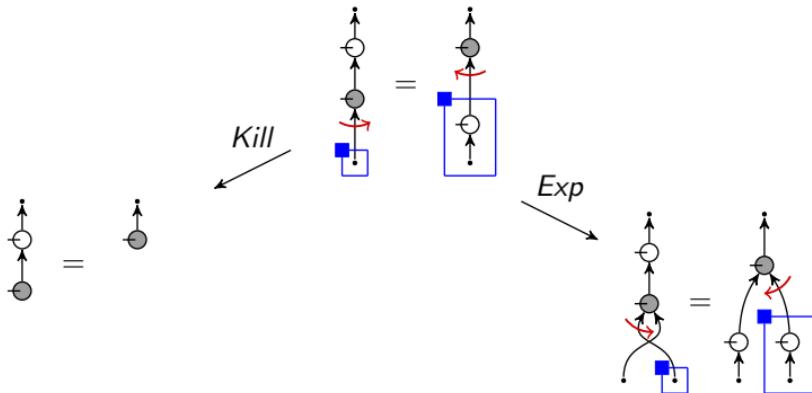
Families of (Non-Commutative) Diagrams



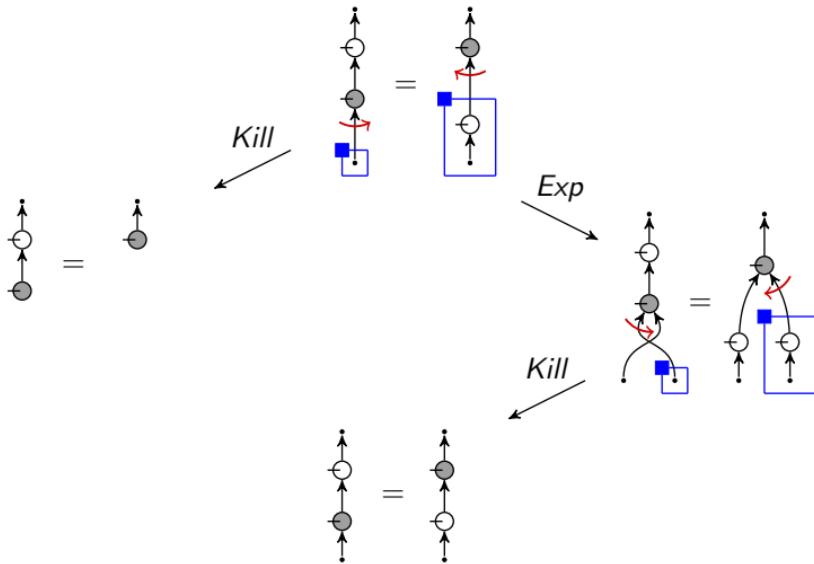
Families of (Non-Commutative) Diagrams



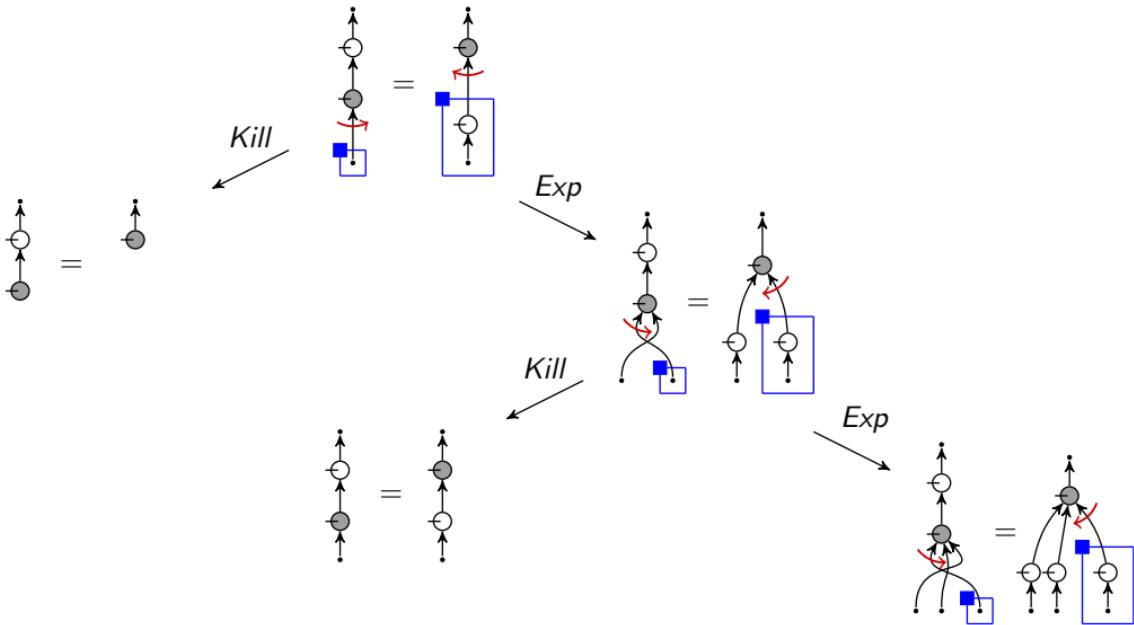
Families of (Non-Commutative) Diagrams



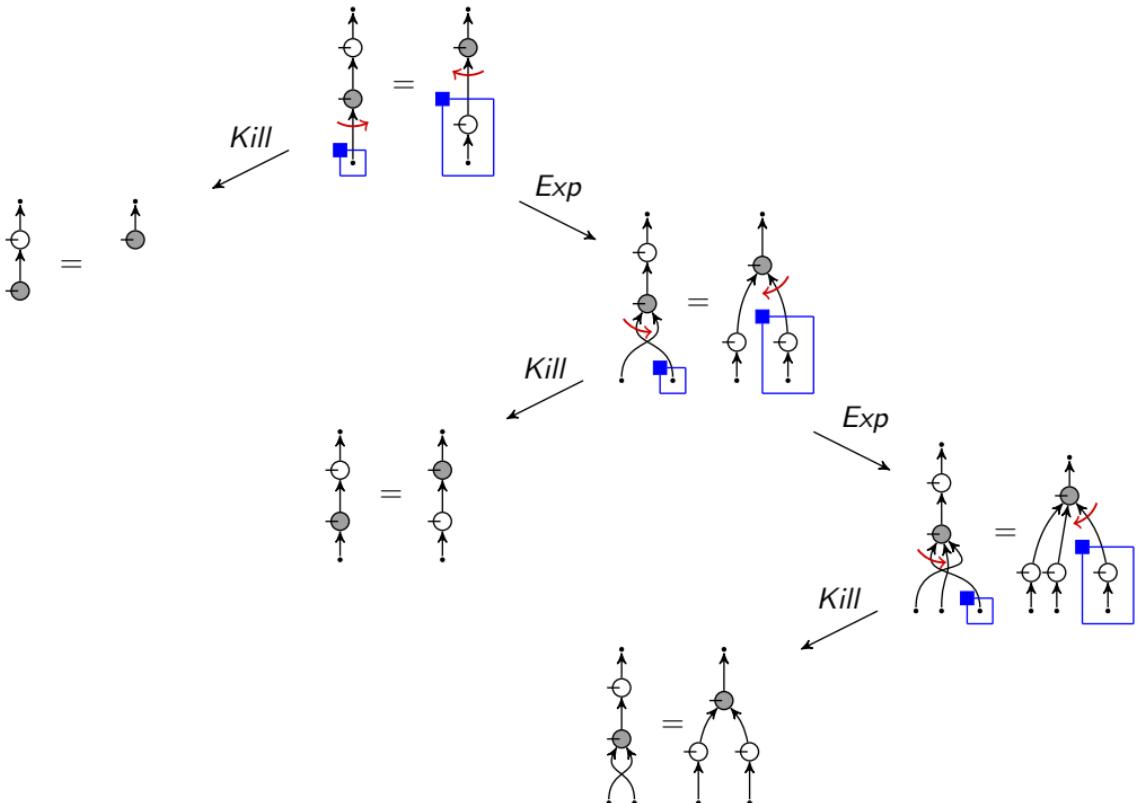
Families of (Non-Commutative) Diagrams



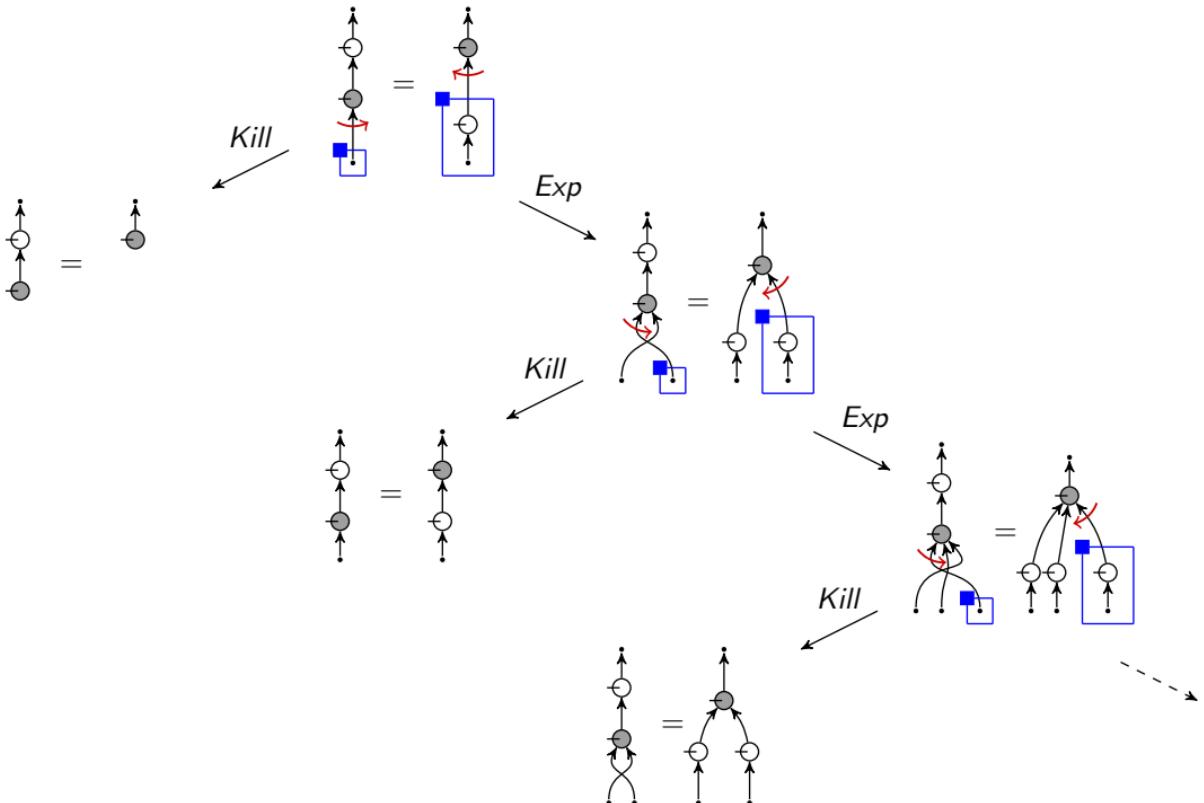
Families of (Non-Commutative) Diagrams



Families of (Non-Commutative) Diagrams

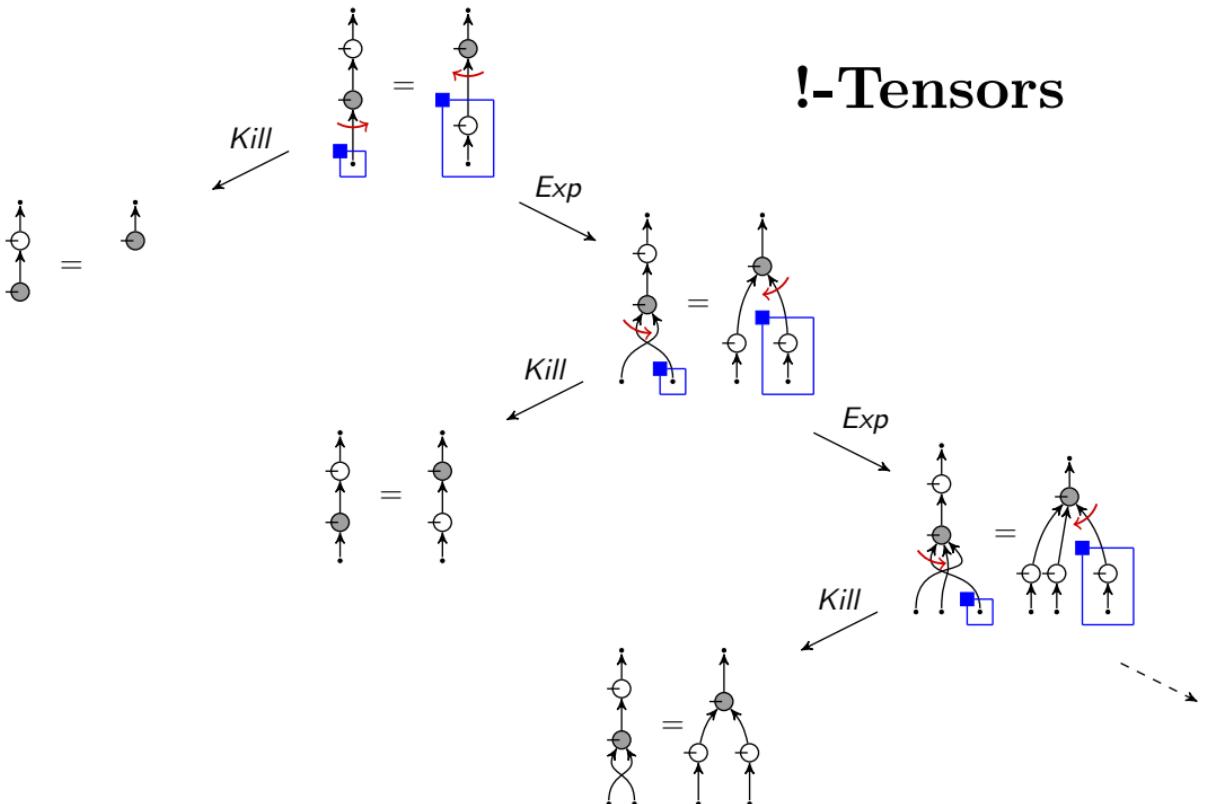


Families of (Non-Commutative) Diagrams



Families of (Non-Commutative) Diagrams

!-Tensors



Quantomatic

QuantoDerive

File Edit Derive Window Export

bialgebra-project-1.0

- axioms
 - distribute.qrule
 - green-id.qrule
 - green-merge.qrule
 - red-id.qrule
 - red-merge.qrule
- derivations
 - bialg-graph.qderive*
- graphs
- simprocs
- theorems

(root)
distribute-0
green-merge-1
green-merge-2
green-id-0
green-id-1
(head)
green-merge-3
red-merge-1

Core status: OK

distribute.qrule x bialg-graph.qderive*

Rewrite Simplify

axioms/distribute (0/0)
axioms/green-id (0/0)
axioms/green-merge (1/1)
axioms/red-id (0/0)
axioms/red-merge (1/2)
theorems/example (0/0)

+ - ↘ ↗ Apply

green-id-1 (head)

Problem/Solution

!-Graphs

!-Tensors

Problem/Solution

!-Graphs

Requires Commutativity

!-Tensors

Problem/Solution

!-Graphs

Requires Commutativity

!-Tensors

Allows Non-Commutativity

Problem/Solution

!-Graphs

Requires Commutativity

Combinatoric

!-Tensors

Allows Non-Commutativity

Problem/Solution

!-Graphs

Requires Commutativity

Combinatoric

!-Tensors

Allows Non-Commutativity

Syntactic

Problem/Solution

!-Graphs

Requires Commutativity

Combinatoric

Automated (Quantomatic)

!-Tensors

Allows Non-Commutativity

Syntactic

Problem/Solution

!-Graphs

Requires Commutativity

Combinatoric

Automated (Quantomatic)

!-Tensors

Allows Non-Commutativity

Syntactic

Not automated

Problem/Solution

!-Graphs

Requires Commutativity

Combinatoric

Automated (Quantomatic)

!-Tensors

Allows Non-Commutativity

Syntactic

Not automated



Problem/Solution

!-Graphs

Requires Commutativity

Combinatoric

Automated (Quantomatic)

!-Tensors

Allows Non-Commutativity

Syntactic

Not automated



Solution 1: Rebuild from scratch

Problem/Solution

!-Graphs

Requires Commutativity

Combinatoric

Automated (Quantomatic)

Solution 1: Rebuild from scratch

!-Tensors

Allows Non-Commutativity

Syntactic

Not automated



Problem/Solution

!-Graphs

Requires Commutativity

Combinatoric

Automated (Quantomatic)

Solution 1: Rebuild from scratch

!-Tensors

Allows Non-Commutativity

Syntactic

Not automated



Solution 2: Adapt Quantomatic

Problem/Solution

!-Graphs

Requires Commutativity

Combinatoric

Automated (Quantomatic)

Solution 1: Rebuild from scratch

!-Tensors

Allows Non-Commutativity

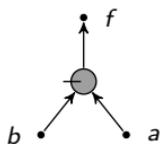
Syntactic

Not automated



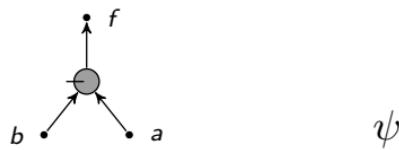
Solution 2: Adapt Quantomatic

!-Tensor Expressions



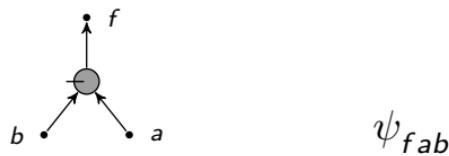
$\psi := \bullet$
$\phi := \circ$

!-Tensor Expressions



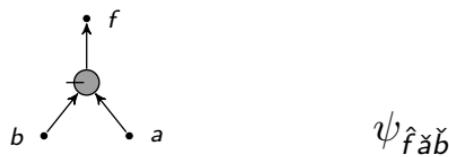
$\psi := \bullet$
$\phi := \circ$

!-Tensor Expressions



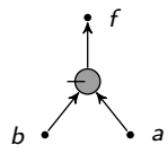
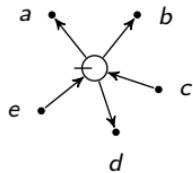
$\psi := \bullet$
 $\phi := \circ$

!-Tensor Expressions



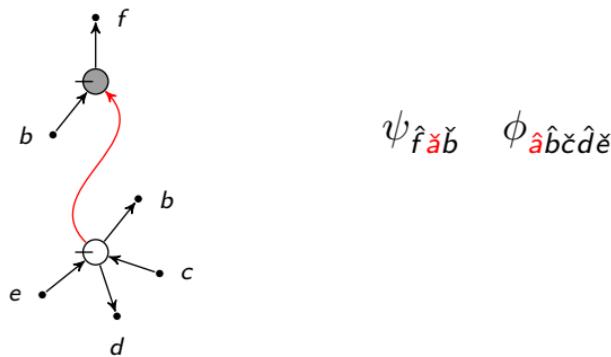
$\psi := \bullet$
$\phi := \circ$

!-Tensor Expressions

 $\psi_{\hat{f} \check{a} \check{b}} \quad \phi_{\hat{a} \hat{b} \check{c} \hat{d} \check{e}}$ 

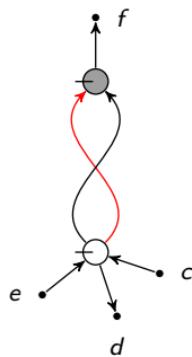
$\psi := \bullet$
$\phi := \circ$

!-Tensor Expressions



$\psi := \bullet$
$\phi := \circ$

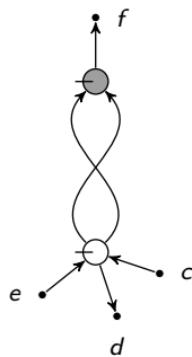
!-Tensor Expressions



$$\psi_{\hat{f}\check{a}\check{b}} \quad \phi_{\hat{a}\hat{b}\check{c}\hat{d}\check{e}}$$

$\psi := \bullet$
$\phi := \circ$

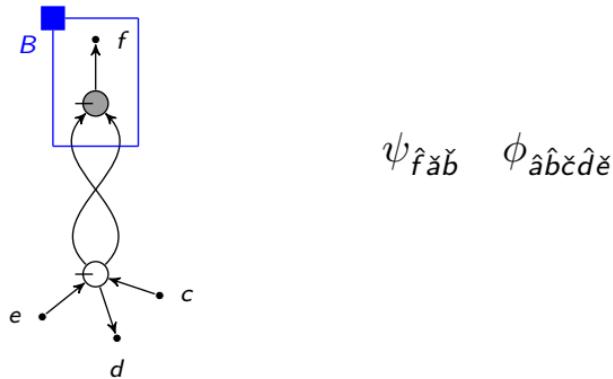
!-Tensor Expressions



$$\psi_{\hat{f}\check{a}\check{b}} \quad \phi_{\hat{a}\hat{b}\check{c}\hat{d}\check{e}}$$

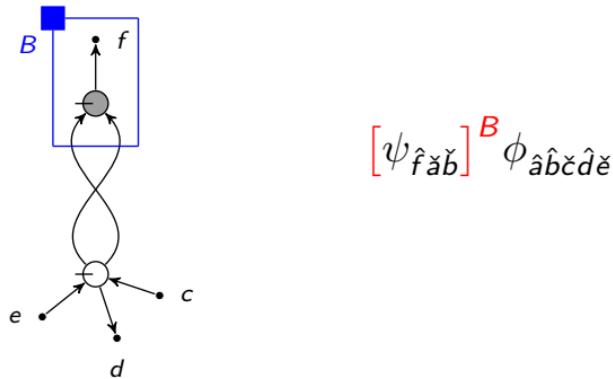
$\psi := \bullet$
$\phi := \circ$

!-Tensor Expressions



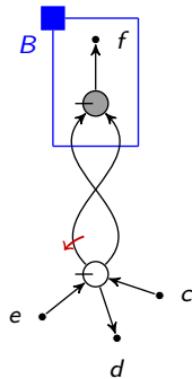
$\psi := \bullet$
$\phi := \circ$

!-Tensor Expressions



$\psi := \bullet$
$\phi := \circ$

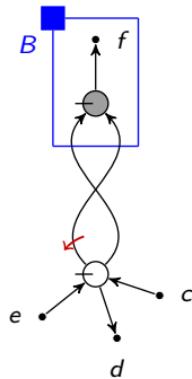
!-Tensor Expressions



$$[\psi_{\hat{f}\check{a}\check{b}}]^B \phi_{\hat{a}\hat{b}\check{c}\hat{d}\check{e}}$$

$$\boxed{\begin{aligned}\psi &:= \bullet \\ \phi &:= \circ\end{aligned}}$$

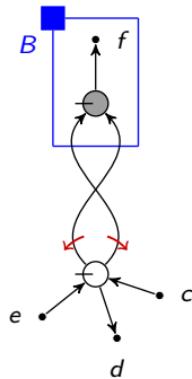
!-Tensor Expressions



$$[\psi_{\hat{f}\check{a}\check{b}}]^B \phi_{\langle \hat{a}]^B \hat{b} \check{c} \hat{d} \check{e}}$$

$$\boxed{\begin{aligned}\psi &:= \bullet \\ \phi &:= \circ\end{aligned}}$$

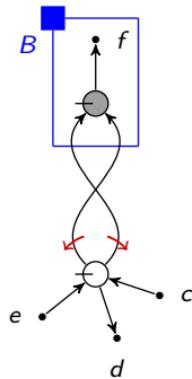
!-Tensor Expressions



$$[\psi_{\hat{f}\check{a}\check{b}}]^B \phi_{\langle \hat{a}]^B \hat{b} \check{c} \hat{d} \check{e}}$$

$$\boxed{\begin{aligned}\psi &:= \bullet \\ \phi &:= \circ\end{aligned}}$$

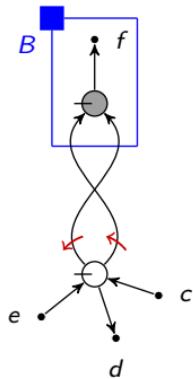
!-Tensor Expressions



$$[\psi_{\hat{f}\check{a}\check{b}}]^B \phi_{\langle \hat{a}]^B [\hat{b})^B \check{c}\hat{d}\check{e}}$$

$\psi := \bullet$
$\phi := \circ$

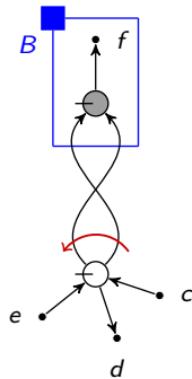
!-Tensor Expressions



$$[\psi_{\hat{f}\check{a}\check{b}}]^B \phi_{\langle \hat{a}]^B \langle \hat{b}]^B \check{c} \hat{d} \check{e}}$$

$\psi := \bullet$
$\phi := \circ$

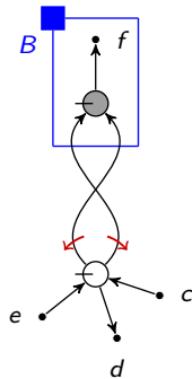
!-Tensor Expressions



$$[\psi_{\hat{f}\check{a}\check{b}}]^B \phi_{\langle \hat{a}\hat{b}]^B \check{c}\hat{d}\check{e}}$$

$\psi := \bullet$
$\phi := \circ$

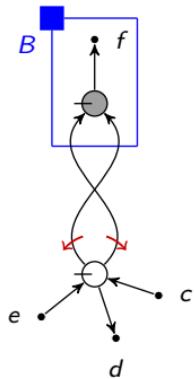
!-Tensor Expressions



$$[\psi_{\hat{f} \check{a} \check{b}}]^B \phi_{\langle \hat{a}]^B [\hat{b})^B \check{c} \hat{d} \check{e}}$$

$\psi := \bullet$
$\phi := \circ$

!-Tensor Expressions



!-Tensor Expression
 $\overbrace{[\psi_{\hat{f}\check{a}\check{b}}]^B \phi_{\langle \hat{a}]^B [\hat{b})^B \check{c}\hat{d}\check{e}}}$

$\psi := \bullet$
$\phi := \circ$

Introduction
○○○○

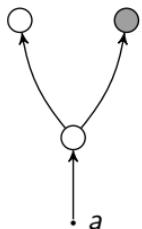
!-Tensors
○

!-Graphs (+ordering)
●○

Isomorphism
○○

Summary
○

!-Graphs



Introduction
○○○○

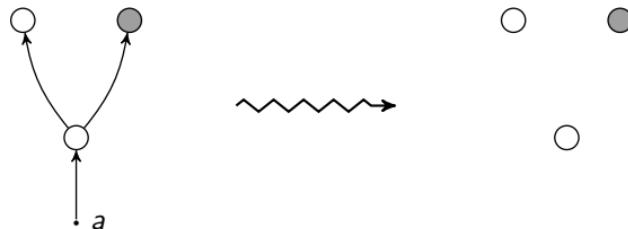
!-Tensors
○

!-Graphs (+ordering)
●○

Isomorphism
○○

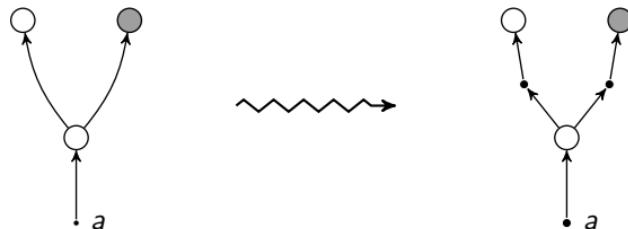
Summary
○

!-Graphs



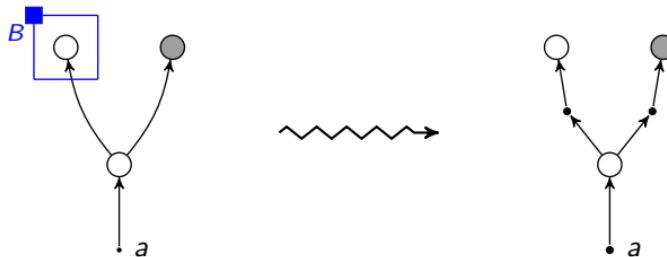
Node-vertices : $\{\text{●}, \text{○}\}$

!-Graphs



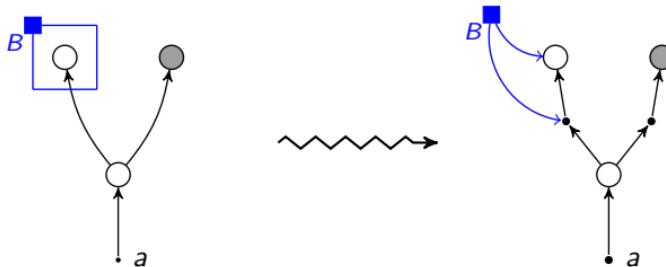
Node-vertices : $\{\bullet, \circ\}$
Wire-vertices : $\{\cdot\}$

!-Graphs



Node-vertices : $\{\bullet, \circ\}$
Wire-vertices : $\{\cdot\}$

!-Graphs

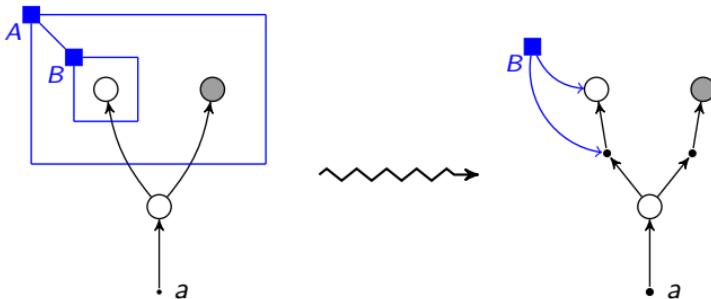


Node-vertices : $\{\text{●}, \text{○}\}$

Wire-vertices : $\{\text{•}\}$

Box-vertices : $\{\blacksquare\}$

!-Graphs

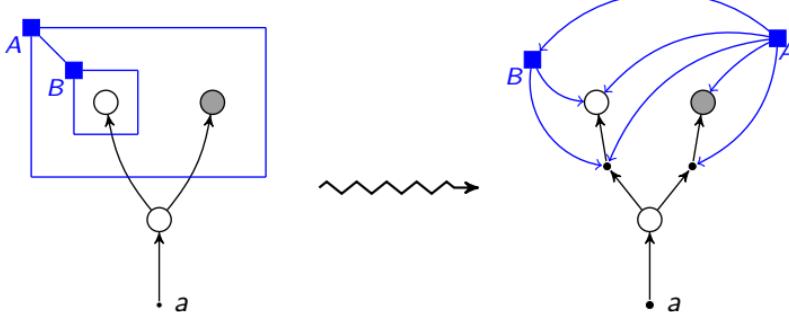


Node-vertices : $\{\bullet, \circ\}$

Wire-vertices : $\{\cdot\}$

Box-vertices : $\{\blacksquare\}$

!-Graphs

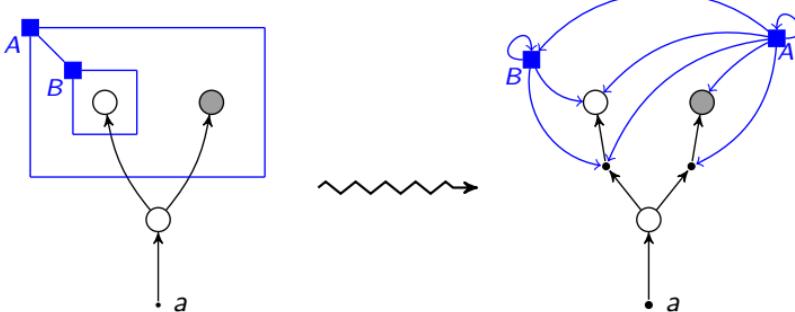


Node-vertices : $\{\bullet, \circ\}$

Wire-vertices : $\{\cdot\}$

Box-vertices : $\{\blacksquare\}$

!-Graphs

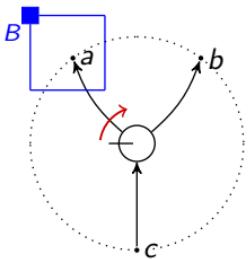


Node-vertices : $\{\bullet, \circ\}$

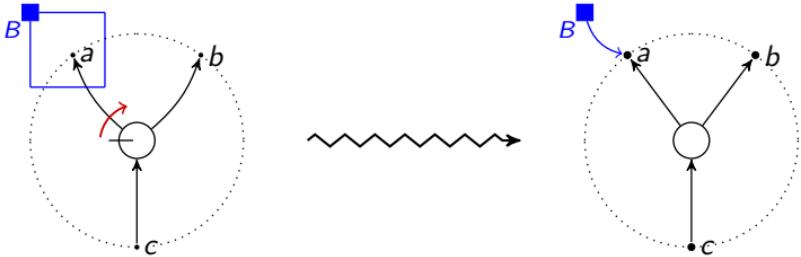
Wire-vertices : $\{\cdot\}$

Box-vertices : $\{\blacksquare\}$

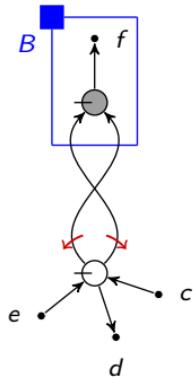
Neighbourhood Orders



Neighbourhood Orders



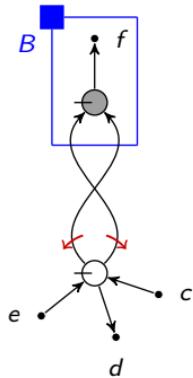
Neighbourhood Orders



!-Tensor Expression
 $\overbrace{[\psi_{\hat{f}\check{a}\check{b}}]^B \phi_{\langle \hat{a}]^B [\hat{b})^B \check{c}\hat{d}\check{e}}}$

$$\begin{array}{l} \psi := \bullet \\ \phi := \circ \end{array}$$

Neighbourhood Orders

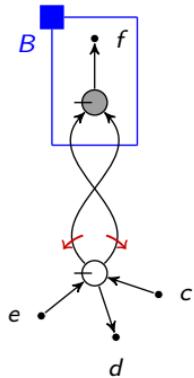


!-Tensor Expression

$$\overbrace{[\psi \hat{f} \check{a} \check{b}]^B}^{\text{!-Tensor Expression}} \phi \underbrace{(\hat{a})^B (\hat{b})^B}_{\text{!-Graphs (+ordering)}} \check{c} \check{d} \check{e}$$

$$\begin{array}{l} \psi := \bullet \\ \phi := \circ \end{array}$$

Neighbourhood Orders



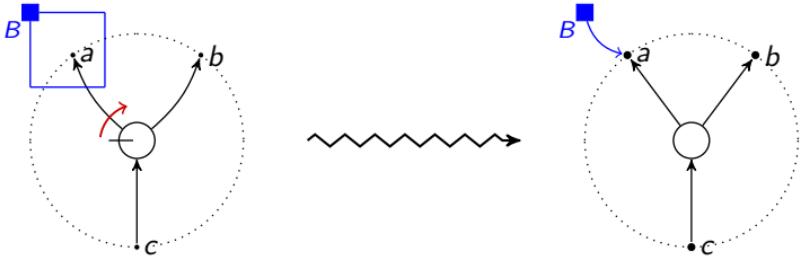
!-Tensor Expression

$$\overbrace{[\psi \hat{f} \check{a} \check{b}]^B \phi \underbrace{\langle \hat{a}]^B [\hat{b})^B \check{c} \hat{d} \check{e}}_{\text{Edgeterms}}}$$

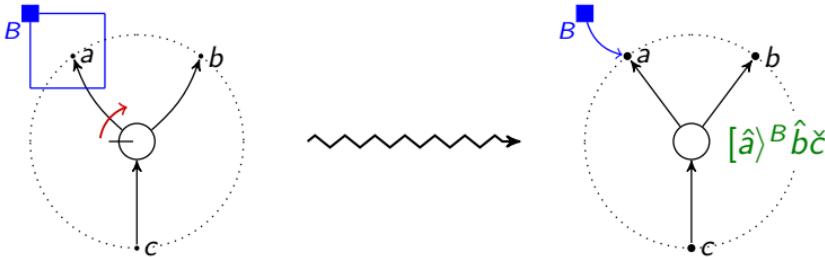
Edgeterms

$$\boxed{\begin{aligned}\psi &:= \bullet \\ \phi &:= \circ\end{aligned}}$$

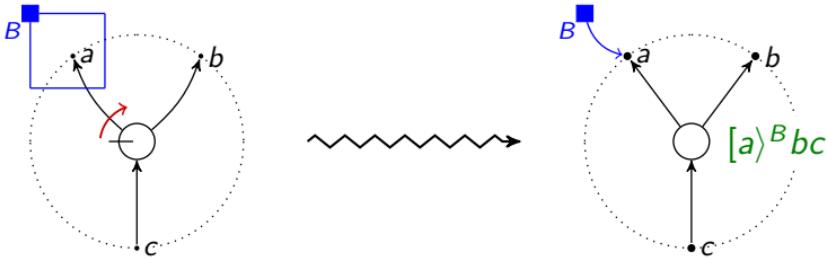
Neighbourhood Orders



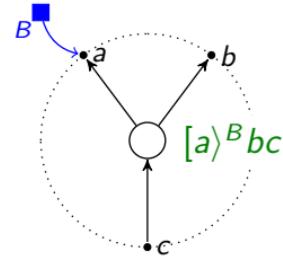
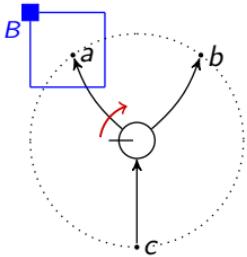
Neighbourhood Orders



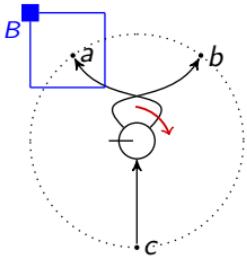
Neighbourhood Orders



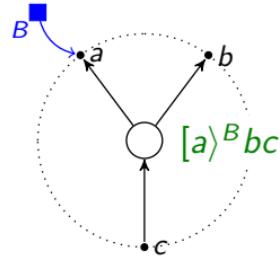
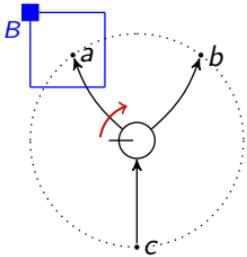
Neighbourhood Orders



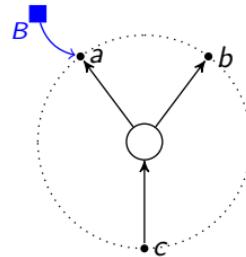
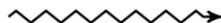
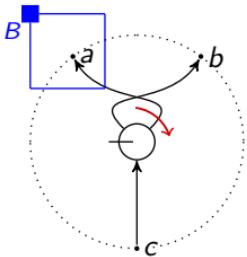
|||



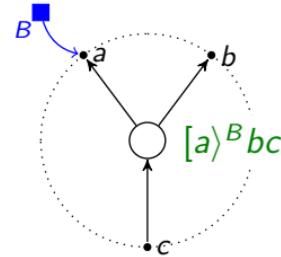
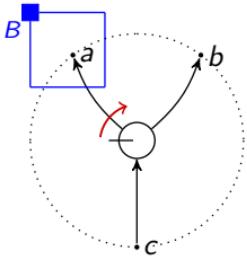
Neighbourhood Orders



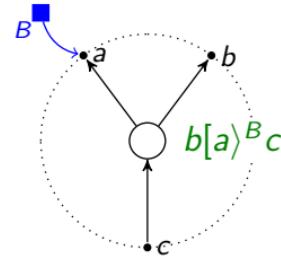
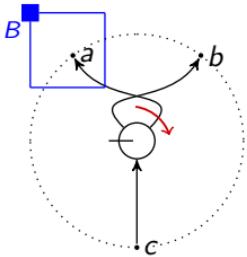
|||



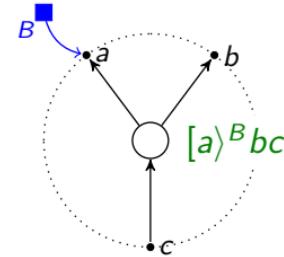
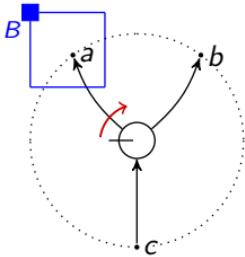
Neighbourhood Orders



|||

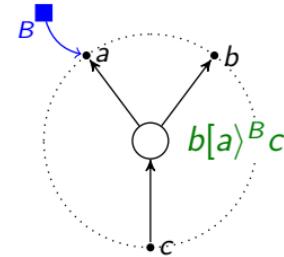
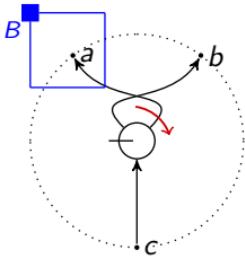


Neighbourhood Orders

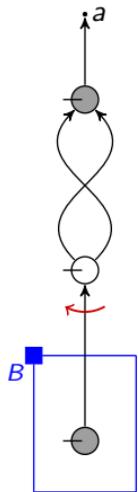


|||

|||

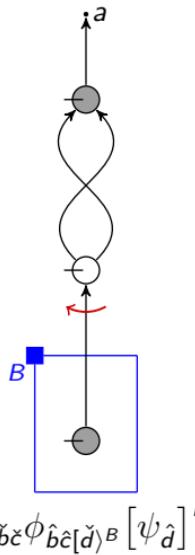


Isomorphism



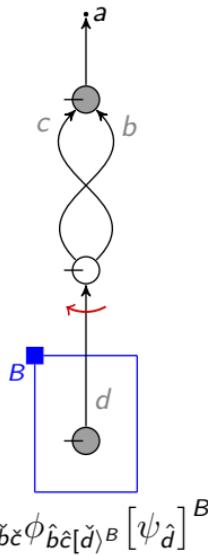
$\psi := \text{---}$
 $\phi := \text{○}$

Isomorphism



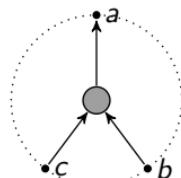
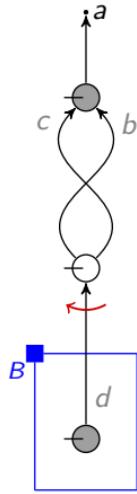
$\psi := \text{---}$
 $\phi := \text{○○}$

Isomorphism



$\psi :=$
$\phi :=$

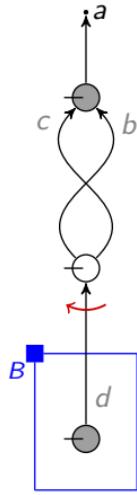
Isomorphism



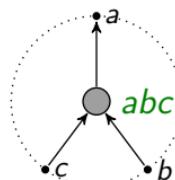
$$\psi_{\hat{a}\check{b}\check{c}} \phi_{\hat{b}\hat{c}} [\check{d}]^B [\psi_{\hat{d}}]^B$$

$\psi := \bullet$
 $\phi := \circ$

Isomorphism

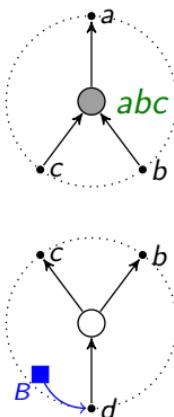
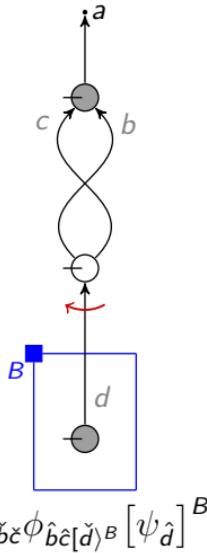


$$\psi_{\hat{a}\check{b}\check{c}} \phi_{\hat{b}\hat{c}} [\check{d}]^B [\psi_{\hat{d}}]^B$$



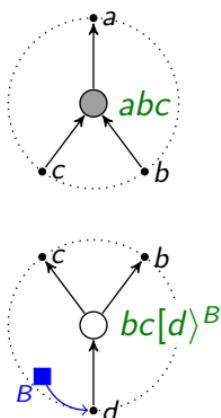
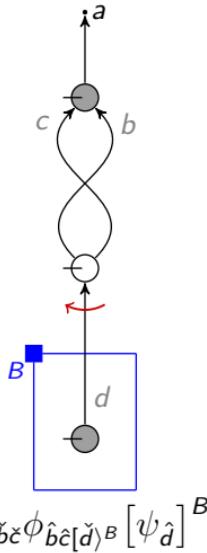
$\psi := \bullet$
$\phi := \circ$

Isomorphism



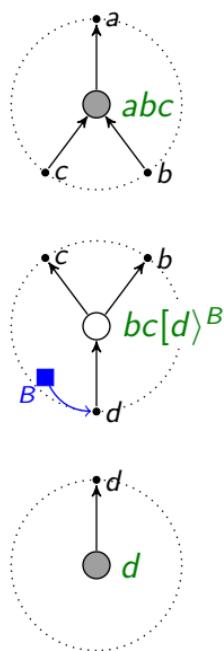
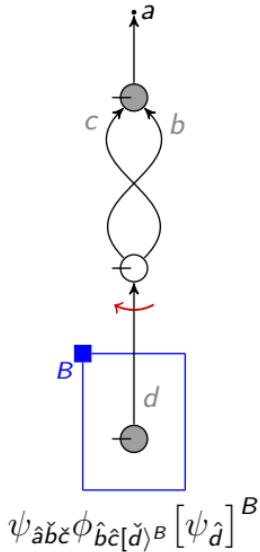
$\psi := \text{---}$
$\phi := \text{○}$

Isomorphism



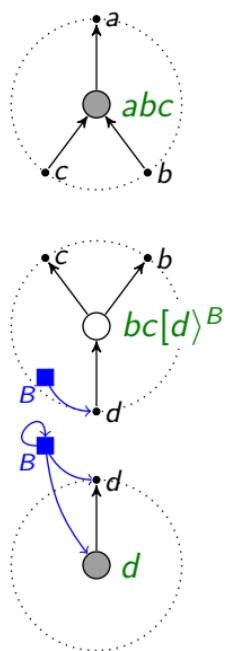
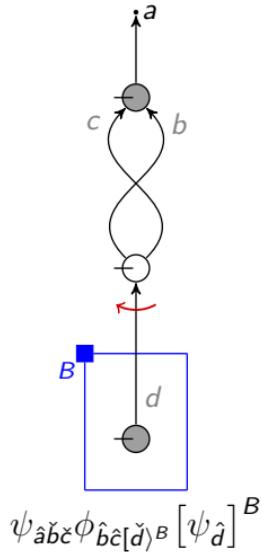
$\psi := \text{---}$
$\phi := \text{○}$

Isomorphism



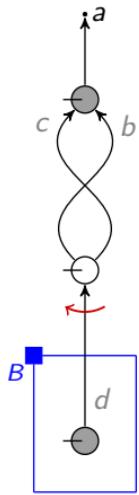
$\psi := \text{---}$
$\phi := \text{○}$

Isomorphism

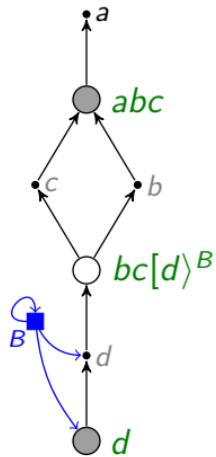


$\psi := \textcircled{-}$
$\phi := \textcircled{\ominus}$

Isomorphism

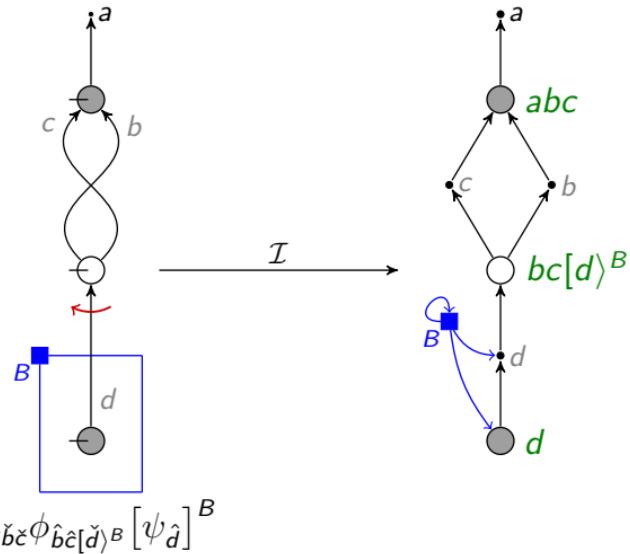


$$\psi_{\hat{a}\check{b}\check{c}} \phi_{\hat{b}\check{c}[\check{d}]^B} [\psi_{\hat{d}}]^B$$



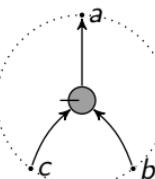
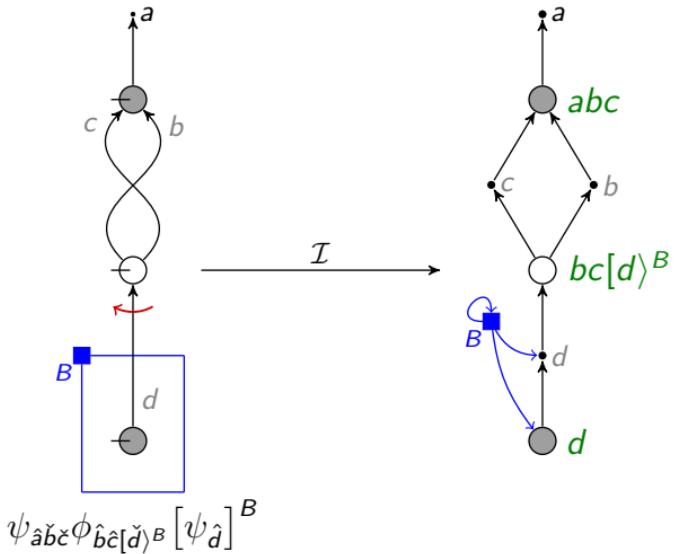
$\psi := \text{---}$
 $\phi := \circ$

Isomorphism



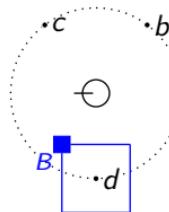
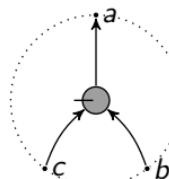
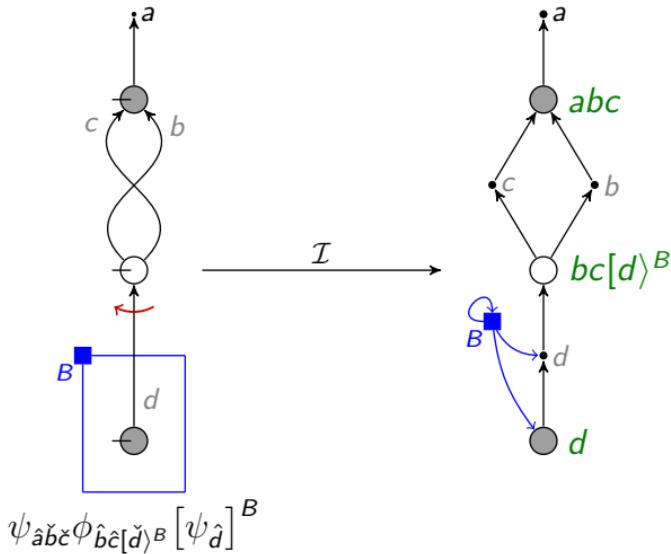
$\psi := \textcircled{-}$
$\phi := \textcircled{\ominus}$

Isomorphism



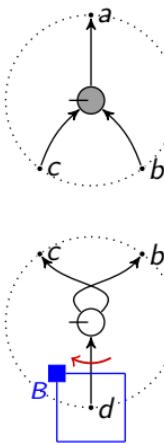
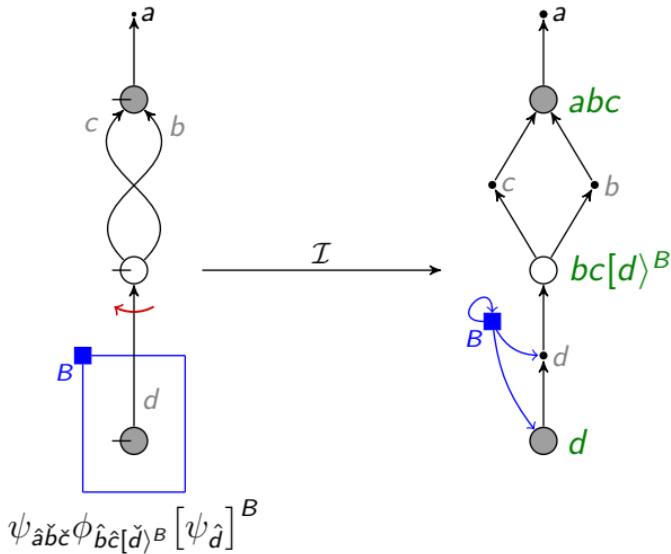
$\psi := \textcircled{\textstyle \ominus}$
$\phi := \textcircled{\textstyle \ominus}$

Isomorphism



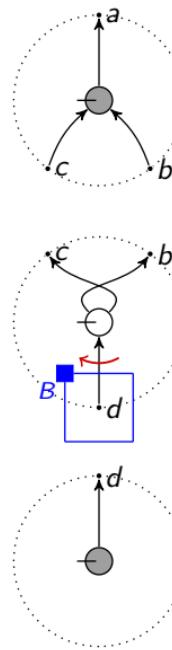
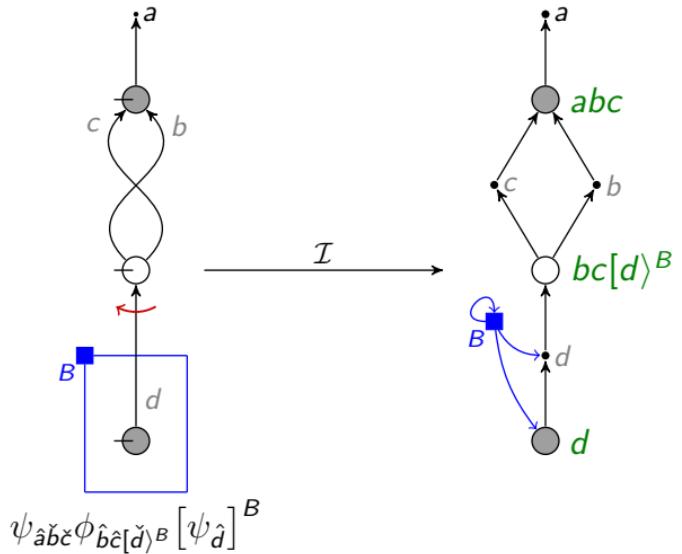
$\psi := \bullet$
$\phi := \circ$

Isomorphism



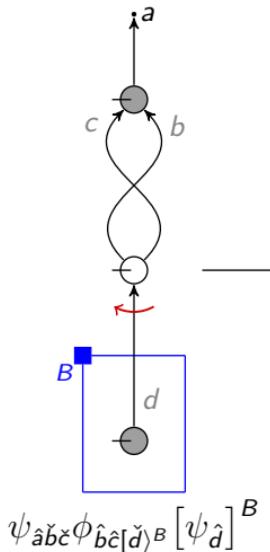
$\psi := \bullet$
$\phi := \circ$

Isomorphism

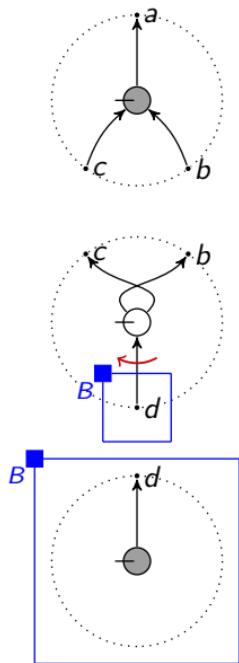
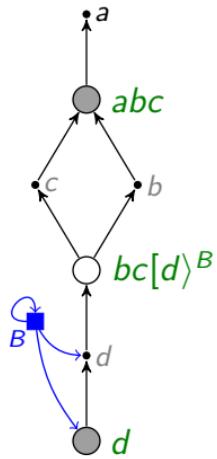


$\psi := \textcircled{-}$
$\phi := \textcircled{\ominus}$

Isomorphism

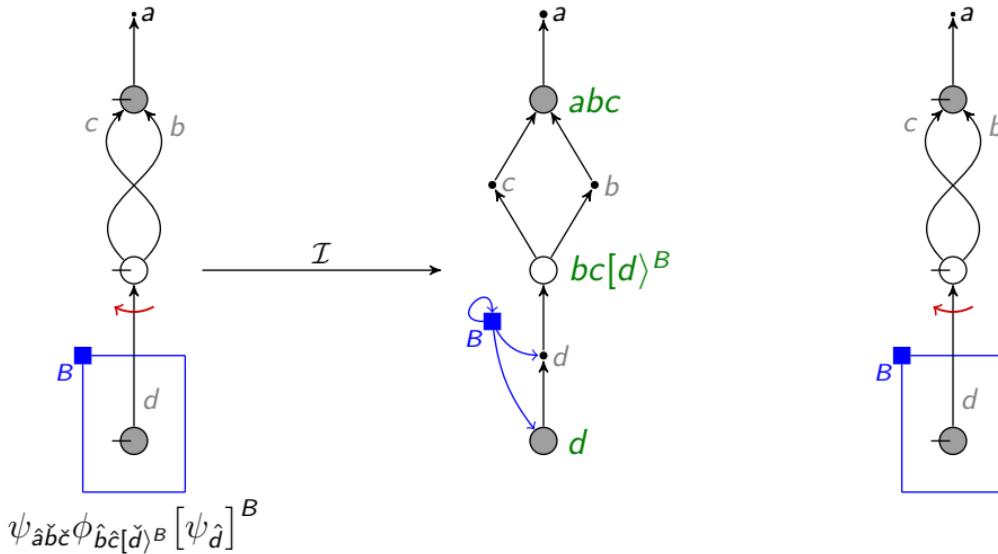


$\xrightarrow{\mathcal{I}}$



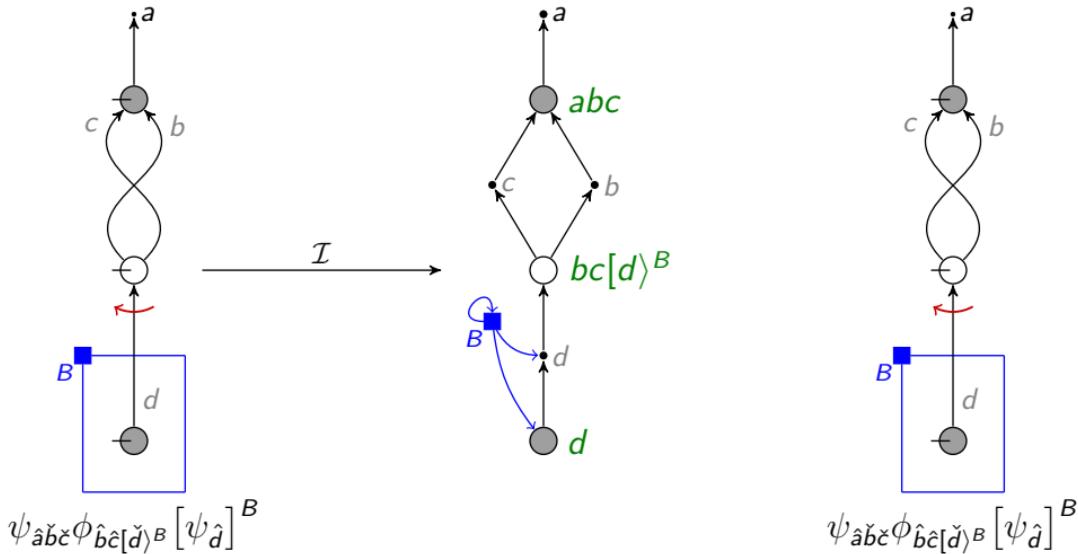
$\psi := \text{---}$
$\phi := \text{○}$

Isomorphism



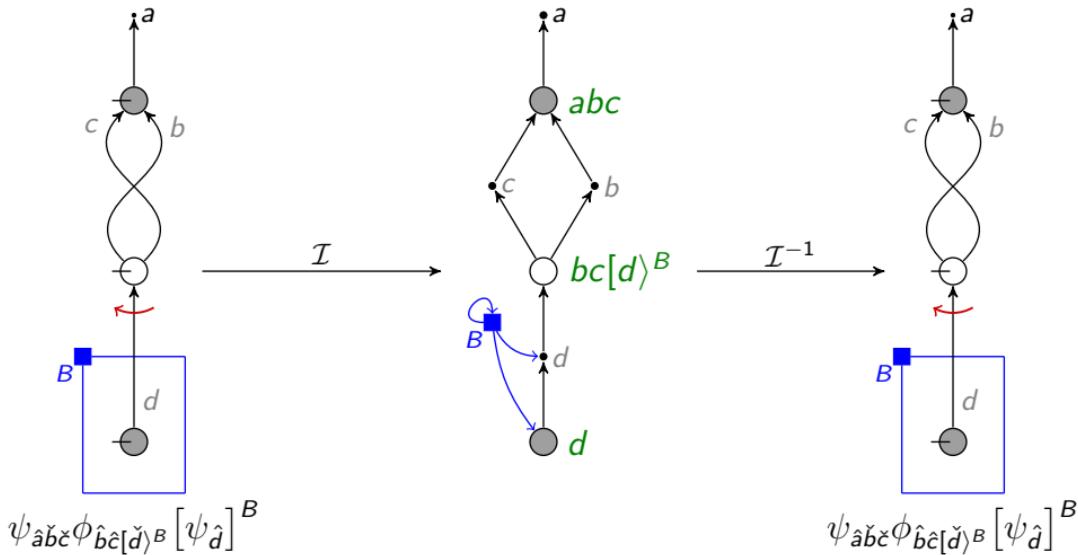
$\psi := \text{---}$
$\phi := \text{○}$

Isomorphism



$\psi := \text{---}$
$\phi := \text{○}$

Isomorphism



$\psi := \bullet$
$\phi := \circ$

Introduction
○○○○○

!-Tensors
○

!-Graphs (+ordering)
○○

Isomorphism
○●

Summary
○

Properties

Properties

$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$

Properties

$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$

Theorem

For G, H two !-tensor expressions, $G \equiv H \implies \mathcal{I}(G) \equiv \mathcal{I}(H)$

Properties

$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$

Theorem

For G, H two !-tensor expressions, $G \equiv H \implies \mathcal{I}(G) \equiv \mathcal{I}(H)$

$\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs + nhd-orders}$

Properties

$$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$$

Theorem

For G, H two !-tensor expressions, $G \equiv H \iff \mathcal{I}(G) \equiv \mathcal{I}(H)$

$$\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs + nhd-orders}$$

Properties

$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$

Theorem

For G, H two !-tensor expressions, $G \equiv H \iff \mathcal{I}(G) \equiv \mathcal{I}(H)$

$\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs + nhd-orders}$

with a left inverse \mathcal{I}^{-1} .

Properties

$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$

Theorem

For G, H two !-tensor expressions, $G \equiv H \iff \mathcal{I}(G) \equiv \mathcal{I}(H)$

$\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs + nhd-orders}$

with a left inverse \mathcal{I}^{-1} .

Theorem

$$\text{!-Tensors} \xrightarrow{\mathcal{I}} \text{!-Graphs + nhd-orders}$$

Properties

$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$

Theorem

For G, H two !-tensor expressions, $G \equiv H \iff \mathcal{I}(G) \equiv \mathcal{I}(H)$

$\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs + nhd-orders}$

with a left inverse \mathcal{I}^{-1} .

Theorem

$$\begin{array}{ccc} \text{!-Tensors} & \xrightarrow{\mathcal{I}} & \text{!-Graphs +} \\ & \downarrow \text{Op}_B & \text{nhd-orders} \\ \text{!-Tensors} & & \end{array}$$

Properties

$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$

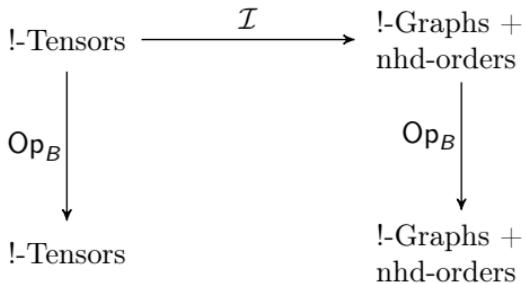
Theorem

For G, H two !-tensor expressions, $G \equiv H \iff \mathcal{I}(G) \equiv \mathcal{I}(H)$

$\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs + nhd-orders}$

with a left inverse \mathcal{I}^{-1} .

Theorem



Properties

$$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$$

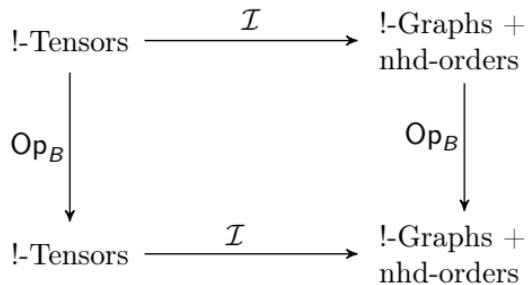
Theorem

For G, H two !-tensor expressions, $G \equiv H \iff \mathcal{I}(G) \equiv \mathcal{I}(H)$

$$\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs + nhd-orders}$$

with a left inverse \mathcal{I}^{-1} .

Theorem



Properties

$$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$$

Theorem

For G, H two !-tensor expressions, $G \equiv H \iff \mathcal{I}(G) \equiv \mathcal{I}(H)$

$$\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs + nhd-orders}$$

with a left inverse \mathcal{I}^{-1} .

Theorem

The following commutes:

$$\begin{array}{ccc} \text{!-Tensors} & \xrightarrow{\mathcal{I}} & \text{!-Graphs +} \\ & \downarrow \text{Op}_B & \downarrow \text{Op}_B \\ \text{!-Tensors} & \xrightarrow{\mathcal{I}} & \text{!-Graphs +} \\ & & \text{nhd-orders} \end{array}$$

Properties

$$\mathcal{I} : \text{!-Tensor expressions} \longrightarrow \text{!-Graphs + nhd-orders}$$

Theorem

For G, H two !-tensor expressions, $G \equiv H \iff \mathcal{I}(G) \equiv \mathcal{I}(H)$

$$\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs + nhd-orders}$$

with a left inverse \mathcal{I}^{-1} .

Theorem

The following commutes:

$$\begin{array}{ccc} \text{!-Tensors} & \xrightarrow{\mathcal{I}} & \text{!-Graphs +} \\ & \downarrow \text{Op}_B & \downarrow \text{Op}_B \\ \text{!-Tensors} & \xleftarrow{\mathcal{I}^{-1}} & \text{!-Graphs +} \\ & & \text{nhd-orders} \end{array}$$

Introduction
○○○○○

!-Tensors
○

!-Graphs (+ordering)
○○

Isomorphism
○○

Summary
●

Summary

Summary

- Quantomatic can't work with !-tensors

Summary

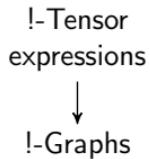
- Quantomatic can't work with !-tensors

!-Tensor
expressions



Summary

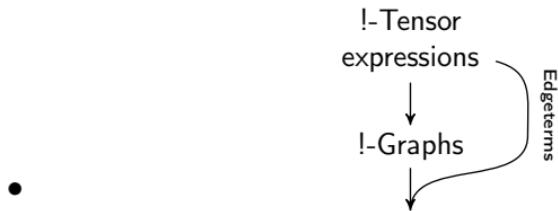
- Quantomatic can't work with !-tensors



•

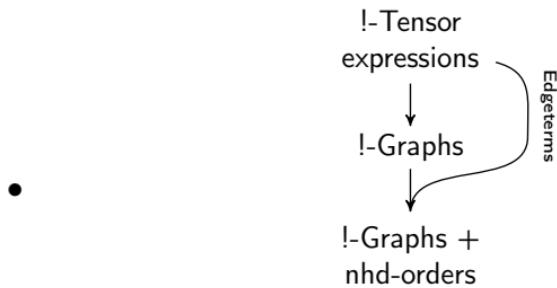
Summary

- Quantomatic can't work with !-tensors



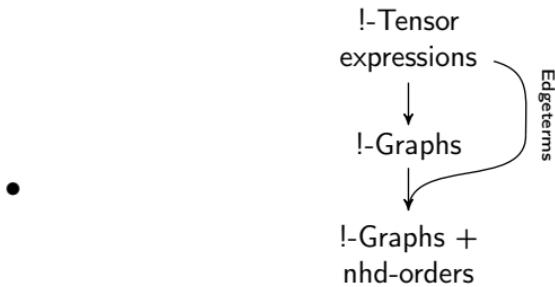
Summary

- Quantomatic can't work with !-tensors



Summary

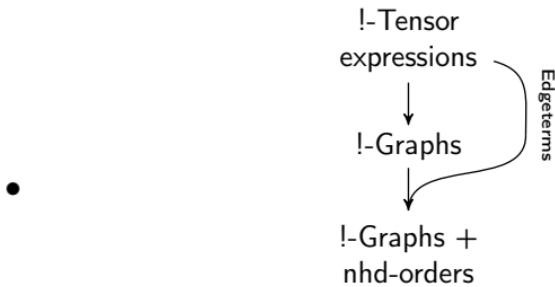
- Quantomatic can't work with !-tensors



- Lift to $\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs} + \text{nhd-orders}$

Summary

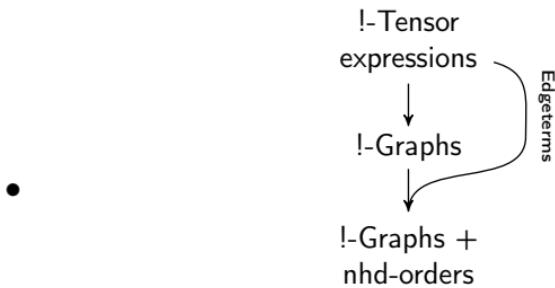
- Quantomatic can't work with !-tensors



- Lift to $\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs} + \text{nhd-orders}$
- Structure preserving

Summary

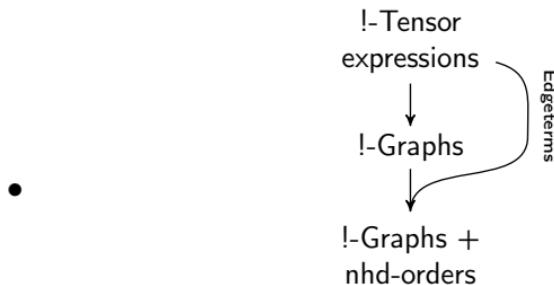
- Quantomatic can't work with !-tensors



- Lift to $\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs} + \text{nhd-orders}$
- Structure preserving
- Encode !-tensors in Quantomatic

Summary

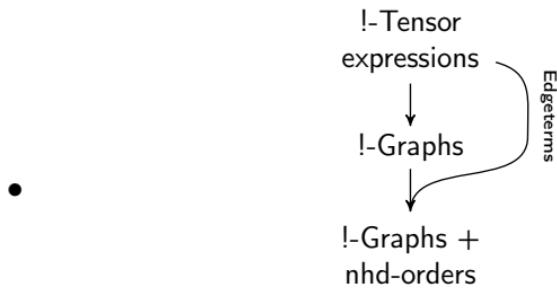
- Quantomatic can't work with !-tensors



- Lift to $\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs} + \text{nhd-orders}$
 - Structure preserving
 - Encode !-tensors in Quantomatic
- ⇒ **Automated reasoning with non-commutative structures**

Summary

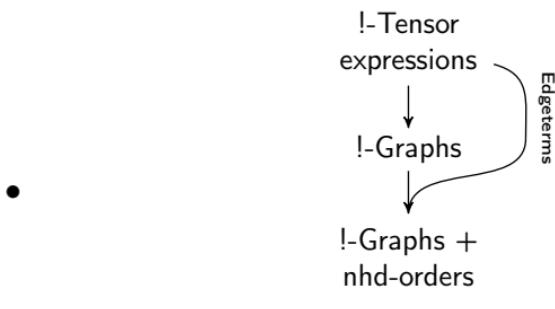
- Quantomatic can't work with !-tensors



- Lift to $\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs} + \text{nhd-orders}$
 - Structure preserving
 - Encode !-tensors in Quantomatic
- ⇒ **Automated reasoning with non-commutative structures**

Summary

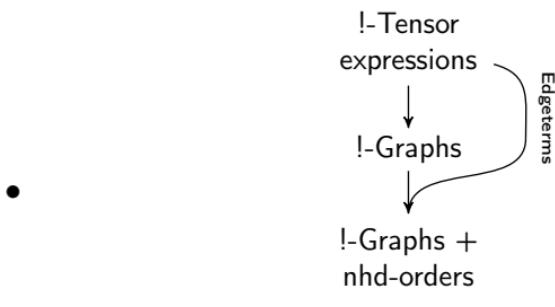
- Quantomatic can't work with !-tensors



- Lift to $\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs} + \text{nhd-orders}$
 - Structure preserving
 - Encode !-tensors in Quantomatic
- ⇒ **Automated reasoning with non-commutative structures**

Summary

- Quantomatic can't work with !-tensors



- Lift to $\mathcal{I} : \text{!-Tensors} \longrightarrow \text{!-Graphs} + \text{nhd-orders}$
 - Structure preserving
 - Encode !-tensors in Quantomatic
- ⇒ **Automated reasoning with non-commutative structures**

Thanks