

An introduction to Graphical Proof Theory

Matteo Acclavio



University of
Southern Denmark

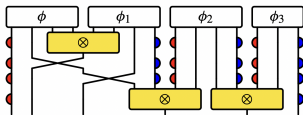
Based (also) on joint works with Ross Horne, Sjouke Mauw, and Lutz Straßburger

Urbino

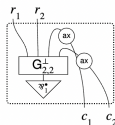
07/09/2023

- 1 Why Graphical Proof Theory?
 - The problem with the 'happens before' relation
- 2 From Graphs to Formulas
 - Modular Decomposition
 - Graphical Connectives
- 3 Graphical proof systems
 - The rules
 - Transitivity of \multimap
 - Conservativity
- 4 Conclusions and Future Works

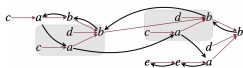
What I do not mean by graphical



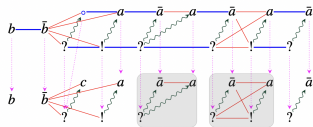
String Diagrams



Generalized Proof Nets



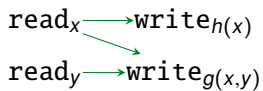
Game Semantics
(Abramsky, Hyland-Ong)



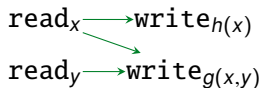
Combinatorial Proofs

Why Graphical Proof Theory?

Happens-before relation is crucial in distributed systems



Happens-before relation is crucial in distributed systems

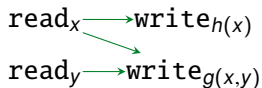


Logical Clocks¹ = Enforcing specific total orders on events

✓	$read_x \triangleleft write_{h(x)} \triangleleft read_y \triangleleft write_{g(x,y)}$
✓	$read_x \triangleleft read_y \triangleleft write_{g(x,y)} \triangleleft write_{h(x)}$
✓	$read_x \triangleleft read_y \triangleleft write_{h(x)} \triangleleft write_{g(x,y)}$
✓	$read_y \triangleleft read_x \triangleleft write_{g(x,y)} \triangleleft write_{h(x)}$
✓	$read_y \triangleleft read_x \triangleleft write_{h(x)} \triangleleft write_{g(x,y)}$
✗	$read_x \triangleleft write_{h(x)} \triangleleft write_{g(x,y)} \triangleleft read_y$

¹Lamport '78

Happens-before relation is crucial in distributed systems



Logical Clocks¹ = Enforcing specific total orders on events

✓	$read_x \triangleleft write_{h(x)} \triangleleft read_y \triangleleft write_{g(x,y)}$
✓	$read_x \triangleleft read_y \triangleleft write_{g(x,y)} \triangleleft write_{h(x)}$
✓	$read_x \triangleleft read_y \triangleleft write_{h(x)} \triangleleft write_{g(x,y)}$
✓	$read_y \triangleleft read_x \triangleleft write_{g(x,y)} \triangleleft write_{h(x)}$
✓	$read_y \triangleleft read_x \triangleleft write_{h(x)} \triangleleft write_{g(x,y)}$
✗	$read_x \triangleleft write_{h(x)} \triangleleft write_{g(x,y)} \triangleleft read_y$

Logical Time = Happens-before relation without clocks

¹Lamport '78

Aim of this line of works:

Proof Theory treating the happens-before relation “logically”

Aim of this line of works:

Proof Theory treating the happens-before relation “logically”

That is:

Logical time is expressed by logical connectives

A “happens before” $B \quad \rightsquigarrow \quad A \triangleleft B$

Aim of this line of works:

Proof Theory treating the happens-before relation “logically”

That is:

Logical time is expressed by logical connectives

A “happens before” B \rightsquigarrow $A \triangleleft B$

and

Logical implication (\dashv) capturing partial order refinements



Pomset logic

Pomset formulas

$$A, B ::= a \mid a^\perp \mid A \wp B \mid A \triangleleft B \mid A \otimes B$$

\wp	\triangleleft	\otimes
disjunction	happens-before	conjunction
parallelism	sequentiality	“independence”

Pomset logic

Pomset formulas

$$A, B ::= a \mid a^\perp \mid A \wp B \mid A \triangleleft B \mid A \otimes B$$

\wp	\triangleleft	\otimes
disjunction	happens-before	conjunction
parallelism	sequentiality	“independence”
(commutative)	(non-commutative)	(commutative)

Pomset formulas

$$A, B ::= a \mid a^\perp \mid A \wp B \mid A \triangleleft B \mid A \otimes B$$

\wp	\triangleleft	\otimes
disjunction	happens-before	conjunction
parallelism	sequentiality	“independence”
(commutative)	(non-commutative)	(commutative)

Negation $(\cdot)^\perp$ such that:

$$A^{\perp\perp} = A$$

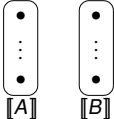
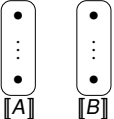
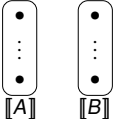
$$(A \wp B)^\perp = A^\perp \otimes B^\perp \quad (A \triangleleft B)^\perp = A^\perp \triangleleft B^\perp \quad (A \otimes B)^\perp = A^\perp \wp B^\perp$$

Implication defined “classically”:

$$A \multimap B := A^\perp \wp B$$

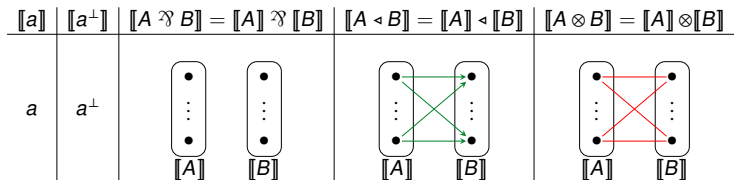
Formulas and Graphs

Relation webs² = graphs encoding Pomset formulas

$\llbracket a \rrbracket$	$\llbracket a^\perp \rrbracket$	$\llbracket A \wp B \rrbracket = \llbracket A \rrbracket \wp \llbracket B \rrbracket$	$\llbracket A \triangleleft B \rrbracket = \llbracket A \rrbracket \triangleleft \llbracket B \rrbracket$	$\llbracket A \otimes B \rrbracket = \llbracket A \rrbracket \otimes \llbracket B \rrbracket$
a	a^\perp			

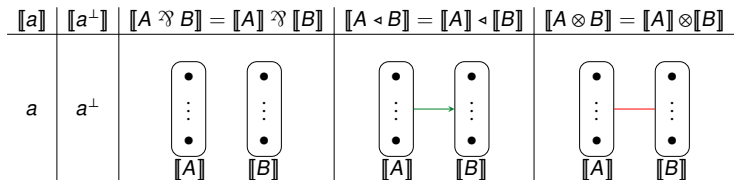
Formulas and Graphs

Relation webs² = graphs encoding Pomset formulas



Formulas and Graphs

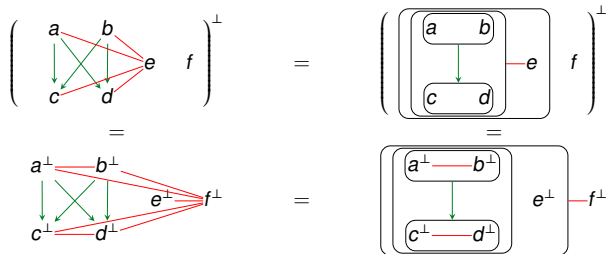
Relation webs² = graphs encoding Pomset formulas



Formulas and Graphs

Relation webs² = graphs encoding Pomset formulas

$[a]$	$[a^\perp]$	$[A \bowtie B] = [A] \bowtie [B]$	$[A \triangleleft B] = [A] \triangleleft [B]$	$[A \otimes B] = [A] \otimes [B]$
a	a^\perp			



Formulas and Graphs

A graph containing an induced subgraph of the following shape cannot be represented by a formula:



or



or



or



or



or



or





Provable in BV

$$(a \not\leq b) \wedge (c \not\leq d) \rightarrow (a \leq c) \wedge (b \leq d)$$



\dashv

Provable in BV

$$(a \wp b) \triangleleft (c \wp d) \dashv (a \triangleleft c) \wp (b \triangleleft d)$$



\dashv

$$(a \wp b) \triangleleft (c \wp d) \dashv \text{NOT A FORMULA}$$



Provable in BV
(series-parallel orders)

$$(a \not\preceq b) \triangleleft (c \not\preceq d) \rightarrow (a \triangleleft c) \not\preceq (b \triangleleft d)$$



$$(a \not\preceq b) \triangleleft (c \not\preceq d) \rightarrow \text{NOT A FORMULA}$$



Provable in BV
(series-parallel orders)

$$(a \not\preceq b) \triangleleft (c \not\preceq d) \dashv\vdash (a \triangleleft c) \not\preceq (b \triangleleft d)$$



Provable in GV^{sl}
(our result)

$$(a \not\preceq b) \triangleleft (c \not\preceq d) \dashv\vdash \text{NOT A FORMULA}$$

Theorem

Let A and B be two **series-parallel** orders.

Then $A \supseteq B$ iff $\vdash_{\text{Pomset}} \{\{A\} \multimap \{B\}\}$

Theorem

The logic BV is a (proper) sub-logic of Pomset such that

If A and B are SP orders, then $A \supseteq B$ iff $\vdash_{\text{BV}} \{\{A\} \multimap \{B\}\}$

Theorem

Let A and B be two **series-parallel** orders.

Then $A \supseteq B$ iff $\vdash_{\text{Pomset}} \{\{A\} \multimap \{B\}\}$

Theorem

The logic BV is a (proper) sub-logic of Pomset such that

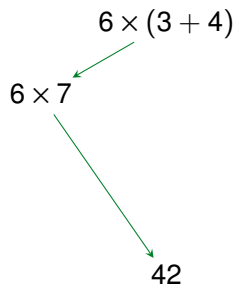
If A and B are SP orders, then $A \supseteq B$ iff $\vdash_{\text{BV}} \{\{A\} \multimap \{B\}\}$

Logic	BV	Pomset
Formalism for proofs	Deep Inference	Proof nets
Complexity	NP	coNP


!!! no possible cut-free sequent calculus for BV or Pomset !!!

What is Deep Inference?

$$6 \times (3 + 4)$$

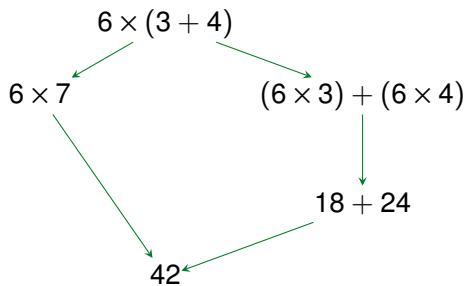


$$6 \times (3 + 4)$$


$$(6 \times 3) + (6 \times 4)$$


$$18 + 24$$


$$42$$



$$\begin{array}{r}
 6 \times (3 + 4) \\
 \hline
 6 \times \left(\underset{\text{sum}}{\frac{3 + 4}{7}} \right) \\
 \hline
 6 \times 7 \\
 \hline
 \underset{\text{mult}}{42}
 \end{array}$$

$$\begin{array}{r}
 6 \times (3 + 4) \\
 \hline
 \underset{\text{dist}}{\left(\begin{array}{r} 6 \times 3 \\ \hline 3 \times 6 \\ \hline \underset{\text{mult}}{18} \end{array} \right) + \left(\begin{array}{r} 6 \times 4 \\ \hline \underset{\text{mult}}{24} \end{array} \right)} \\
 \hline
 \underset{\text{sum}}{\frac{18 + 24}{42}}
 \end{array}$$

Deep inference does the same!

$$\bullet G \implies \begin{array}{c} G \\ \text{Id}_G \parallel \\ G \end{array} \left(\text{or } \begin{array}{c} G' \\ \cdots \\ G \end{array} \text{ if } G = G' \right)$$

Deep inference does the same!

- $G \implies \text{Id}_G \parallel G \quad \left(\text{or } \begin{array}{c} G' \\ \cdots \\ G \end{array} \text{ if } G = G' \right)$

- $\begin{array}{c} H_1 \\ \mathcal{D}_1 \parallel \\ G_1 \end{array} \text{ and } \begin{array}{c} H_2 \\ \mathcal{D}_2 \parallel \\ G_2 \end{array} \text{ and } \text{rule} \frac{G_1}{H_2} \implies \text{rule} \frac{\begin{array}{c} H_1 \\ \mathcal{D}_1 \parallel \\ G_1 \end{array}}{\begin{array}{c} H_2 \\ \mathcal{D}_2 \parallel \\ G_2 \end{array}}$

Deep inference does the same!

- $G \implies \text{Id}_G \parallel \begin{array}{c} G \\ G \end{array} \quad \left(\text{or } \begin{array}{c} G' \\ \cdots \\ G \end{array} \text{ if } G = G' \right)$

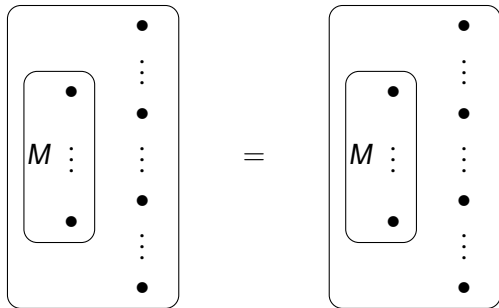
- $\begin{array}{c} H_1 \\ \mathcal{D}_1 \parallel \\ G_1 \end{array} \text{ and } \begin{array}{c} H_2 \\ \mathcal{D}_2 \parallel \\ G_2 \end{array} \text{ and } \text{rule} \frac{G_1}{H_2} \implies \text{rule} \frac{\begin{array}{c} H_1 \\ \mathcal{D}_1 \parallel \\ G_1 \end{array}}{\begin{array}{c} H_2 \\ \mathcal{D}_2 \parallel \\ G_2 \end{array}}$

- $\begin{array}{c} H_i \\ \mathcal{D}_i \parallel \\ G_i \end{array} \text{ and } P \text{ an } n\text{-ary term-constructor} \implies P \left(\begin{array}{c} H_1 \\ \mathcal{D}_1 \parallel \\ G_1 \end{array}, \dots, \begin{array}{c} H_n \\ \mathcal{D}_n \parallel \\ G_n \end{array} \right)$

From Graphs to Formulas

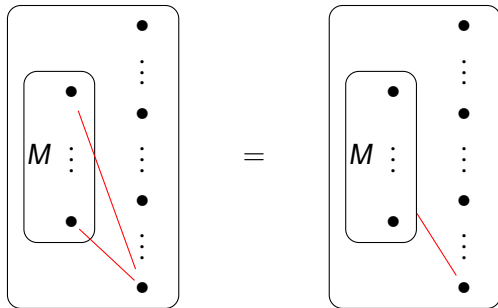
Modular Decomposition

A **module** of a graph $G = H[M]$ is a set of vertices M s.t.



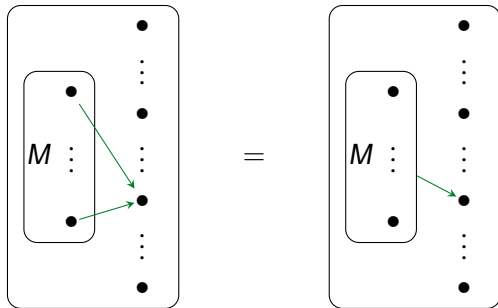
Modular Decomposition

A **module** of a graph $G = H[M]$ is a set of vertices M s.t.



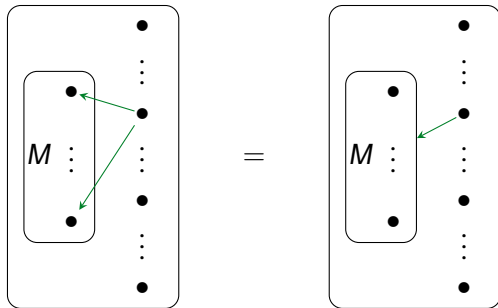
Modular Decomposition

A **module** of a graph $G = H[M]$ is a set of vertices M s.t.



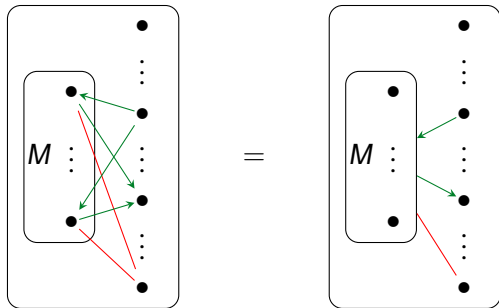
Modular Decomposition

A **module** of a graph $G = H[M]$ is a set of vertices M s.t.



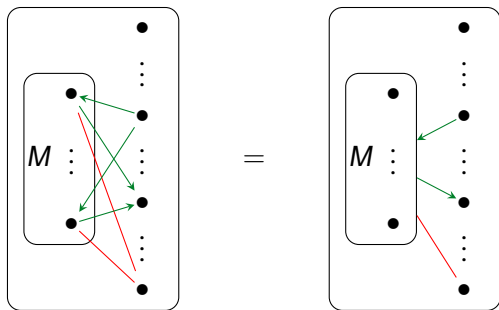
Modular Decomposition

A **module** of a graph $G = H[M]$ is a set of vertices M s.t.

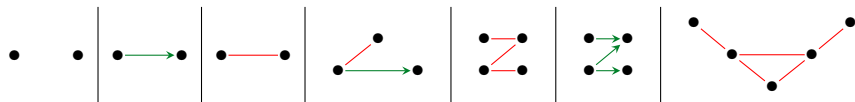


Modular Decomposition

A **module** of a graph $G = H[M]$ is a set of vertices M s.t.



A graph G is **prime** if it has modules V_G , \emptyset and $\{x\}$ for all $x \in V_G$.



If G has n vertices and H_1, \dots, H_n graphs,
 then we use G as a **logic connective** and we write $G(H_1, \dots, H_n)$

$$\wp: \begin{array}{c} \bullet \quad \bullet \\ \wp(G, H) = G \wp H \end{array} \quad \left| \quad \triangleleft: \begin{array}{c} \bullet \xrightarrow{\text{green}} \bullet \\ \triangleleft(G, H) = G \triangleleft H \end{array} \quad \left| \quad \otimes: \begin{array}{c} \bullet \xrightarrow{\text{red}} \bullet \\ \otimes(G, H) = G \otimes H \end{array} \right.$$

If G has n vertices and H_1, \dots, H_n graphs,
 then we use G as a **logic connective** and we write $G(H_1, \dots, H_n)$

$$\wp: \begin{array}{c} \bullet \quad \bullet \\ \wp(G, H) = G \wp H \end{array} \quad \left| \quad \triangleleft: \begin{array}{c} \bullet \xrightarrow{\text{green}} \bullet \\ \triangleleft(G, H) = G \triangleleft H \end{array} \quad \left| \quad \otimes: \begin{array}{c} \bullet \xrightarrow{\text{red}} \bullet \\ \otimes(G, H) = G \otimes H \end{array} \right.$$

Lemma (Modular decomposition of graphs (Gallai '75))

If $G \neq \emptyset$ is a graph, then we have exactly one of the following cases:

- (i) G is a singleton graph
- (ii) $G = P(A_1, \dots, A_n)$ for a prime graph P

Graphical proof systems

The rules

$$\text{ai}\downarrow \frac{\emptyset}{a^\perp \wp a}$$

$$\text{ai}\uparrow \frac{a^\perp \otimes a}{\emptyset}$$

$$\text{s}\wp \frac{P(M_1, \dots, M_{i-1}, \mathbf{M}_i \wp N, M_{i+1}, \dots, M_n)}{\mathbf{M}_i \wp P(M_1, \dots, M_{i-1}, N, M_{i+1}, \dots, M_n)}$$

$$\text{s}\otimes \frac{\mathbf{M}_i \otimes P(M_1, \dots, M_{i-1}, N, M_{i+1}, \dots, M_n)}{P(M_1, \dots, M_{i-1}, \mathbf{M}_i \otimes N, M_{i+1}, \dots, M_n)}$$

$$\text{p}\downarrow \frac{(\mathbf{M}_1 \wp N_1) \otimes \dots \otimes (\mathbf{M}_n \wp N_n)}{R^\perp(\mathbf{M}_1, \dots, \mathbf{M}_n) \wp R(N_1, \dots, N_n)}$$

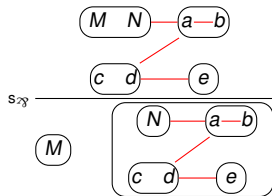
$$\text{p}\uparrow \frac{R(\mathbf{M}_1, \dots, \mathbf{M}_n) \otimes R^\perp(N_1, \dots, N_n)}{(\mathbf{M}_1 \otimes N_1) \wp \dots \wp (\mathbf{M}_n \otimes N_n)}$$

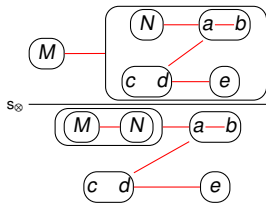
$$\text{q}\downarrow \frac{Q^\perp(\mathbf{M}_1 \wp N_1, \dots, \mathbf{M}_n \wp N_n)}{Q^\perp(\mathbf{M}_1, \dots, \mathbf{M}_n) \wp Q(N_1, \dots, N_n)}$$

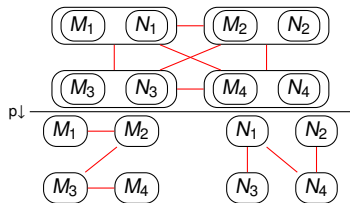
$$\text{q}\uparrow \frac{Q(\mathbf{M}_1, \dots, \mathbf{M}_n) \otimes Q^\perp(N_1, \dots, N_n)}{Q(\mathbf{M}_1 \otimes N_1, \dots, \mathbf{M}_n \otimes N_n)}$$

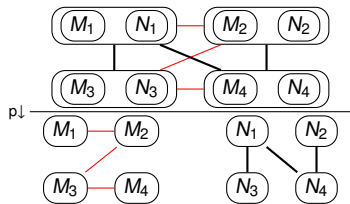
$$\text{qm} \frac{Q(\mathbf{M}_1 \wp N_1, \dots, \mathbf{M}_n \wp N_n)}{Q(\mathbf{M}_1, \dots, \mathbf{M}_n) \wp Q(N_1, \dots, N_n)}$$

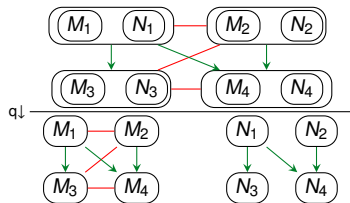
$$\text{sl} \frac{Q(M_1, \dots, M_k, \emptyset, \dots, \emptyset) \triangleleft Q(\emptyset, \dots, \emptyset, M_{k+1}, \dots, M_n)}{Q(M_1, \dots, M_n)}$$

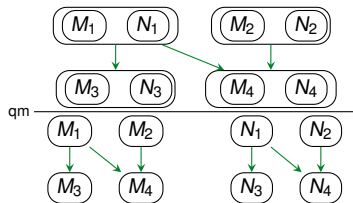




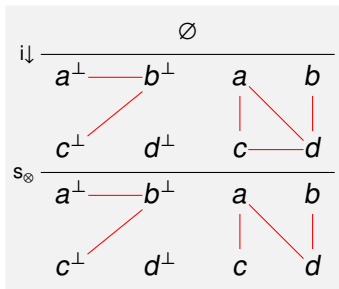




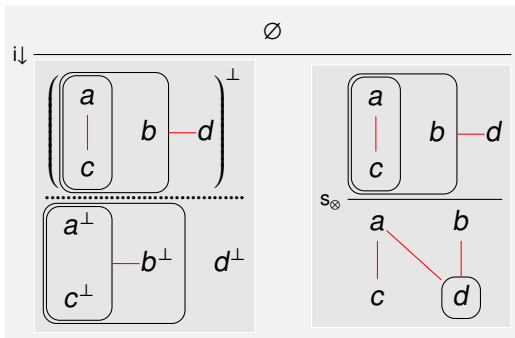




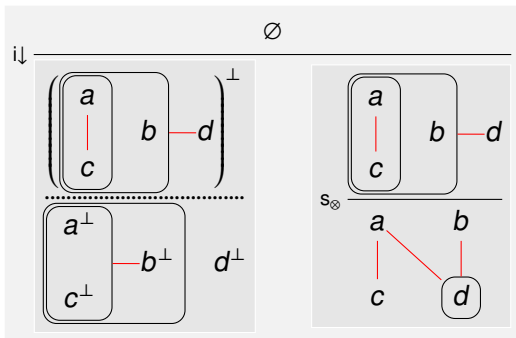
An example:



An example:

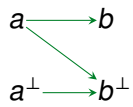


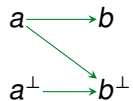
An example:



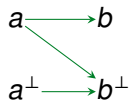
That is:

$$\vdash_{\text{GS}} \begin{array}{cc} a & b \\ | & | \\ c & -d \end{array} \quad \dashv \quad \begin{array}{cc} a & b \\ | & | \\ c & d \end{array}$$

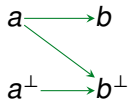




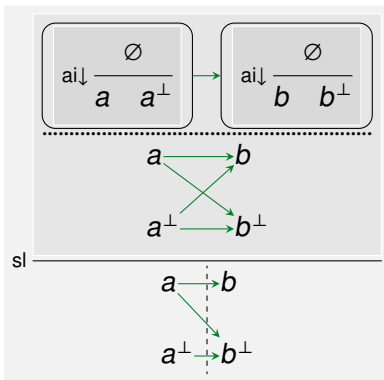
The rule sl refines a partial order *slicing* a “before” and an “after”



The rule sl refines a partial order *slicing* a “before” and an “after”
This rule implements Logical Clocks!



The rule *sl* refines a partial order *slicing* a “before” and an “after”
 This rule implements Logical Clocks!



Transitivity of \rightarrow

Remark

if $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$

Remark

if $(A^\perp \cong B)$ and $(B^\perp \cong C)$, then $(A^\perp \cong C)$

Remark

if $(A^\perp \cong B) \otimes (B^\perp \cong C)$, then $(A^\perp \cong C)$

Remark

if $(A^\perp \cong B) \otimes (B^\perp \cong C)$, then $(A^\perp \cong C)$

$$\begin{array}{c}
 \begin{array}{cc}
 \emptyset & \emptyset \\
 \parallel & \parallel \\
 A^\perp \cong B & B^\perp \cong C
 \end{array} \otimes \\
 \hline
 \text{p}\downarrow \\
 \begin{array}{c}
 A^\perp \cong \text{i}\uparrow \frac{B^\perp \otimes B}{\emptyset} \cong C \\
 = \\
 A^\perp \cong C
 \end{array}
 \end{array}$$

Remark

if $(A^\perp \wp B) \otimes (B^\perp \wp C)$, then $(A^\perp \wp C)$

$$\begin{array}{c}
 \begin{array}{cc}
 \begin{array}{c} \emptyset \\ \parallel \\ A^\perp \wp B \end{array} & \otimes & \begin{array}{c} \emptyset \\ \parallel \\ B^\perp \wp C \end{array} \\
 \hline
 \text{p}\downarrow \\
 \begin{array}{c}
 A^\perp \wp \begin{array}{c} \text{i}\uparrow \frac{B^\perp \otimes B}{\emptyset} \wp C \\
 \hline
 A^\perp \wp C
 \end{array}
 \end{array}
 \end{array}
 \end{array}$$

Cut Elimination = the rule $\text{i}\uparrow \frac{A \otimes A^\perp}{\emptyset}$ is admissible

Theorem

The rule $i\uparrow$ is derivable in $\{ai\uparrow, p\uparrow, q\uparrow\}$

Theorem

- *rules $\{ai\uparrow, p\uparrow, s_{\otimes}\}$ are admissible in $GS = \{ai\downarrow, s_{\otimes}, p\downarrow\}$*
- *rules $\{ai\uparrow, p\uparrow, q\uparrow\}$ are admissible in $GV = \{ai\downarrow, s_{\otimes}, s_{\otimes}, p\downarrow, q\downarrow, qm\}$*
- *rules $\{ai\uparrow, p\uparrow, q\uparrow\}$ are admissible in $GV^{sl} = \{ai\downarrow, s_{\otimes}, s_{\otimes}, p\downarrow, q\downarrow, qm, sl\}$*

$$\text{ai}\downarrow \frac{\emptyset}{a^\perp \wp a}$$

$$\text{ai}\uparrow \frac{a^\perp \otimes a}{\emptyset}$$

$$\text{s}\wp \frac{P(M_1, \dots, M_{i-1}, \mathbf{M}_i \wp N, M_{i+1}, \dots, M_n)}{\mathbf{M}_i \wp P(M_1, \dots, M_{i-1}, N, M_{i+1}, \dots, M_n)}$$

$$\text{s}\otimes \frac{\mathbf{M}_i \otimes P(M_1, \dots, M_{i-1}, N, M_{i+1}, \dots, M_n)}{P(M_1, \dots, M_{i-1}, \mathbf{M}_i \otimes N, M_{i+1}, \dots, M_n)}$$

$$\text{p}\downarrow \frac{(\mathbf{M}_1 \wp N_1) \otimes \dots \otimes (\mathbf{M}_n \wp N_n)}{R^\perp(\mathbf{M}_1, \dots, \mathbf{M}_n) \wp R(N_1, \dots, N_n)}$$

$$\text{p}\uparrow \frac{R(\mathbf{M}_1, \dots, \mathbf{M}_n) \otimes R^\perp(N_1, \dots, N_n)}{(\mathbf{M}_1 \otimes N_1) \wp \dots \wp (\mathbf{M}_n \otimes N_n)}$$

$$\text{q}\downarrow \frac{Q^\perp(\mathbf{M}_1 \wp N_1, \dots, \mathbf{M}_n \wp N_n)}{Q^\perp(\mathbf{M}_1, \dots, \mathbf{M}_n) \wp Q(N_1, \dots, N_n)}$$

$$\text{q}\uparrow \frac{Q(\mathbf{M}_1, \dots, \mathbf{M}_n) \otimes Q^\perp(N_1, \dots, N_n)}{Q(\mathbf{M}_1 \otimes N_1, \dots, \mathbf{M}_n \otimes N_n)}$$

$$\text{qm} \frac{Q(\mathbf{M}_1 \wp N_1, \dots, \mathbf{M}_n \wp N_n)}{Q(\mathbf{M}_1, \dots, \mathbf{M}_n) \wp Q(N_1, \dots, N_n)}$$

$$\text{sl} \frac{Q(M_1, \dots, M_k, \emptyset, \dots, \emptyset) \triangleleft Q(\emptyset, \dots, \emptyset, M_{k+1}, \dots, M_n)}{Q(M_1, \dots, M_n)}$$

How to prove “cut-elimination”?

How to prove “cut-elimination”?

Splitting:

Pick a connected component of a provable $G \wp P(M_1, \dots, M_n)$.
You can apply rules to G until you have a rule eliminating P .

How to prove “cut-elimination”?

Splitting:

Pick a connected component of a provable $G \wp P(M_1, \dots, M_n)$.
You can apply rules to G until you have a rule eliminating P .

Context-Reduction:

You can apply splitting to a connector P deep in a context.

How to prove “cut-elimination”?

Splitting:

Pick a connected component of a provable $G \wp P(M_1, \dots, M_n)$.
You can apply rules to G until you have a rule eliminating P .

Context-Reduction:

You can apply splitting to a connector P deep in a context.

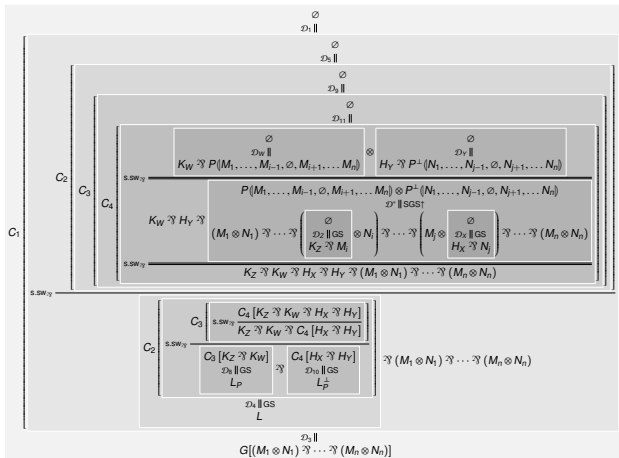
Up-elimination:

If the premise of an up-rule is provable, then its conclusion also is.

Remark: up-rules elimination is not trivial

Remark: up-rules elimination is not trivial

One case of the $p\uparrow$ -elimination in GS:



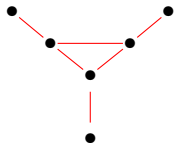
Conservativity

$$\vdash_{GX} \llbracket F \rrbracket \implies \vdash_X F$$

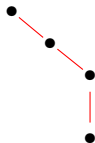
Note: $A \star \emptyset = \star(A, \emptyset) = A$

Note: $A \star \emptyset = \star(A, \emptyset) = A$

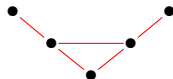
A new notion of “*sub-formula*” analyticity arises from this work:



has sub-connective (e.g.)

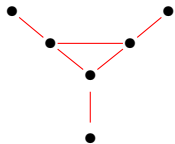


and

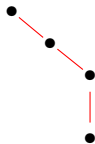


Note: $A \star \emptyset = \star(A, \emptyset) = A$

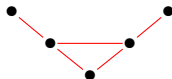
A new notion of “*sub-formula*” analyticity arises from this work:



has sub-connective (e.g.)



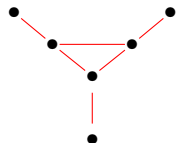
and



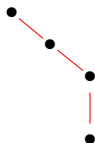
Analytic proof: no “new proper connectives” occurs during proof search

Note: $A \star \emptyset = \star(A, \emptyset) = A$

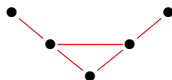
A new notion of “*sub-formula*” analyticity arises from this work:



has sub-connective (e.g.)



and



Analytic proof: no “new proper connectives” occurs during proof search

Theorem

If G is provable, then G admits an analytic proof.

Conclusions and Future Works

Our results:

(-) GS, GV and GV^{sl} are proof systems [in the sense of Cook-Reckhow]

Our results:

- (-) GS, GV and GV^{sl} are proof systems [in the sense of Cook-Reckhow]
- (-) Size of a proof is polynomially bounded

Our results:

- (-) GS, GV and GV^{sl} are proof systems [in the sense of Cook-Reckhow]
- (-) Size of a proof is polynomially bounded
- (-) GS is a conservative extension of MLL°

Our results:

- (-) GS, GV and GV^{sl} are proof systems [in the sense of Cook-Reckhow]
- (-) Size of a proof is polynomially bounded
- (-) GS is a conservative extension of MLL°
- (-) GV and GV^{sl} are both conservative extensions of both BV and GS

Future works:

(-) Categorical and Algebraic Semantics

Future works:

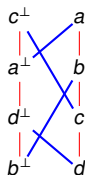
- (-) Categorical and Algebraic Semantics
- (-) Topological correctness criteria:
 - Correctness criterion for GS [on ArXiv!]



Satisfies Retoré's Criterion
Provable in GS



Satisfies Retoré's Criterion
NOT provable in GS



Doesn't satisfy Retoré's Criterion
Provable in GS

- Correctness criterion for BV \implies criteria for GV and GS

Future works:

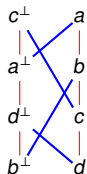
- (-) Categorical and Algebraic Semantics
- (-) Topological correctness criteria:
 - Correctness criterion for GS [on ArXiv!]



Satisfies Retoré's Criterion
Provable in GS



Satisfies Retoré's Criterion
NOT provable in GS



Doesn't satisfy Retoré's Criterion
Provable in GS

- Correctness criterion for BV \implies criteria for GV and GS
- (-) Full homomorphism requires new tools

Future works:

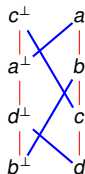
- (-) Categorical and Algebraic Semantics
- (-) Topological correctness criteria:
 - Correctness criterion for GS [on ArXiv!]



Satisfies Retoré's Criterion
Provable in GS



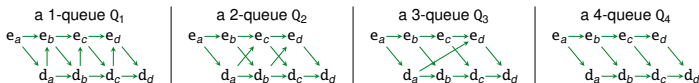
Satisfies Retoré's Criterion
NOT provable in GS



Doesn't satisfy Retoré's Criterion
Provable in GS

- Correctness criterion for BV \implies criteria for GV and GS

- (-) Full homomorphism requires new tools



$\vdash_{BV} Q_1 \dashv\dashv Q_2 \quad \vdash_{GV^{sl}} Q_2 \dashv\dashv Q_3 \quad \vdash_{???} Q_3 \dashv\dashv Q_4$

Future works:

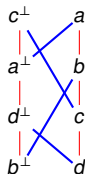
- (-) Categorical and Algebraic Semantics
- (-) Topological correctness criteria:
 - Correctness criterion for GS [on ArXiv!]



Satisfies Retoré's Criterion
Provable in GS



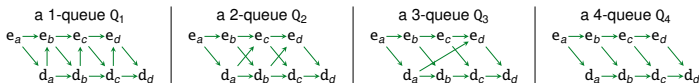
Satisfies Retoré's Criterion
NOT provable in GS



Doesn't satisfy Retoré's Criterion
Provable in GS

- Correctness criterion for BV \implies criteria for GV and GS

- (-) Full homomorphism requires new tools



$$\vdash_{BV} Q_1 \multimap Q_2 \quad \vdash_{GV^{sl}} Q_2 \multimap Q_3 \quad \vdash_{???} Q_3 \multimap Q_4$$

- (-) Applications to Games/Event Structures/Concurrency/...

Thank you

Thank you

Questions?
Comments?