Correctness as Good Interactive Behavior

Master's Thesis

Adrien Ragot, supervised by
Prof. Lorenzo Tortora De Falco

Roma tre University & Aix-Marseille University

June 20, 2021

Some conventions.

Additive Linear Logic   Multiplicative Linear Logic

$$\begin{tikzcd} \oplus & \vee \arrow[l] \arrow[r] & \parr \\ \& & \wedge \arrow[l] \arrow[r] & \otimes \\ \top & T \arrow[l] \arrow[r] & 1 \\ 0 & F \arrow[l] \arrow[r] & \bot \end{tikzcd}$$

What is Logic? Intuitions from linguistics.

In logic, grossly, we seek to tell if a statement is true or false.

Statements = Formulas
A formula is a syntaxical object, a succesion of symbols.

Example :
Socrate is a man, AND,
Socrate is not a man OR Socrate is mortal,
SO, Socrate is mortal.

Statement = Formula = Atomic Formulas + Connectors
The previous statement can be associated to the formula

$$(A \otimes (A^{\perp} \parr B)) \to B$$

.

Towards a notion of proof.

But to the previous statement we can also associate a *proof*.

$$\dfrac{\textit{Socrate is a man} \quad \textit{Socrate is not a man}, \textit{Socrate is mortal}}{\textit{Socrate is mortal}}\ cut$$

Bottom formula = Conclusion. Top Formula = Premisses. (Intuition)
Rules preserve "truths".
A proof is an object that allows us to agree on the validity of a formula (under some **context**).

Gentzen's Sequent Calculus = Proof formalism

A one–sided sequent :

$$A_1, ..., A_n \vdash B_1, ..., B_k$$

It has to be understood as the validity of the formula
$A_1 \otimes ... \otimes A_n \Rightarrow B_1 \parr ... \parr B_k$ Rules = Premisse to Conclusion

Logics

Logic = Formula's + Rules
Classical logic has a **strong symmetry** therefore it allows for one sided sequents:

$$\vdash A_1, ..., A_n$$

Linear Logic =

- refinement of Classical Logic.
- Formulas–as–ressources.
- **Controlling** the *contraction* and *weakening* rules (with the exponentials).

$$\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A}\, c \qquad \frac{\vdash \Gamma}{\vdash \Gamma, A}\, w$$

Why controlling contraction? it is costy.

For proof–search.

$$\dfrac{\vdash \Gamma, A, ..., A}{\vdash \Gamma, A}\; c{\times}n$$

For cut–elimination.

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma, A, A}}{\vdash \Gamma, A}\,c \qquad \dfrac{\pi_2}{\vdash \Delta, \overline{A}}}{\vdash \Gamma, \Delta}\,cut \quad \xrightarrow{cutelim} \quad \dfrac{\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma, A, A} \qquad \dfrac{\pi_2}{\vdash \Delta, \overline{A}}}{\vdash \Gamma, \Delta, A}\,cut \qquad \dfrac{\pi_2}{\vdash \Delta, \overline{A}}}{\vdash \Gamma, \Delta, \Delta}\,cut}{\vdash \Gamma, \Delta}\,c{\times}n$$

Multiplicative Linear Logic (**MLL**)

$$A, B = X \in \mathcal{V} \quad | \quad A^{\perp} \quad | \quad A \parr B \quad | \quad A \otimes B$$

$$(A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp} \qquad (A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp}$$

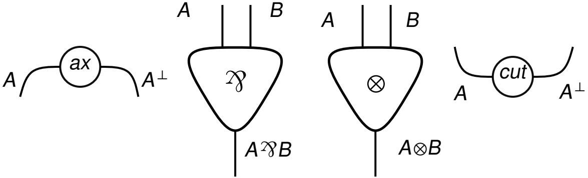
$$\frac{}{\vdash A, \overline{A}}\, ax \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, \overline{A}}{\vdash \Gamma, \Delta}\, cut$$

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\, \otimes \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\, \parr$$

Figure 1: Formulas, de Morgan laws, and rules of the **MLL** fragment

Cells for MLL, Modules, Proof structures

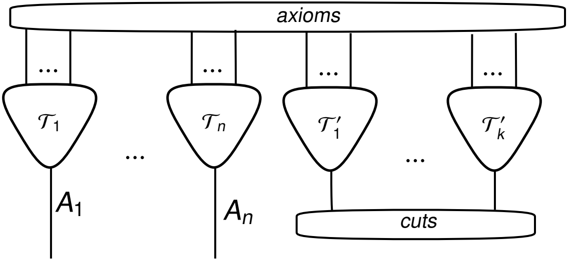
Module = graph constructed the following cells;



Output = arrows with no target. Input = arrows with no source.

Modules = Forest + axioms + cuts

Proof structure = graph constructed the previous cells + No input arrow

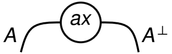


axioms = pairs of dual input arrows
cut = pairs of dual output arrows

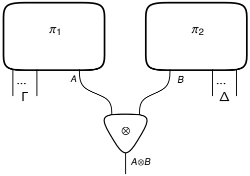
Proof net (1/2)

Proof net = Proof structure representing an MLL–proof.

$$\left[\frac{}{\vdash A, \overline{A}}\,ax\right] =$$

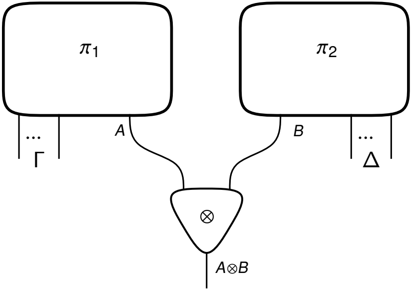


$$\left[\frac{\begin{array}{c}\Vert\,\pi\\ \vdash \Gamma, A, B\end{array}}{\vdash \Gamma, A \parr B}\,\parr\right] =$$

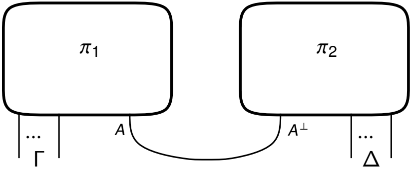


Proof net (2/2)

$$\left[\dfrac{\begin{array}{c}\Big\Vert\,\pi_1\\ \vdash \Gamma, A\end{array} \qquad \begin{array}{c}\Big\Vert\,\pi_2\\ \vdash \Delta, B\end{array}}{\vdash \Gamma, \Delta, A \otimes B}\otimes\right] =$$

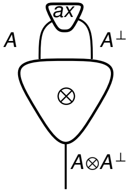


$$\left[\dfrac{\begin{array}{c}\Big\Vert\,\pi_1\\ \vdash \Gamma, A\end{array} \qquad \begin{array}{c}\Big\Vert\,\pi_2\\ \vdash \Delta, \overline{A}\end{array}}{\vdash \Gamma, \Delta}\,cut\right] =$$

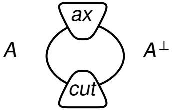


Not all proof structures are proof nets.

This does not represent a proof:



The following structure is called the *deadlock*;



How to characterize proof nets? Notion of switching

One answer : *the correctness criterion* of Danos-Regnier.

Definition (Switching)

A *switching* of a proof structure, is a function σ that chooses for each $\parr$–cell one of the two input.

The switching structure σS is defined as S such that:

- we remove each $\parr$–cells.
- For each $\parr$–cell v,
  - The input $\sigma(v)$ takes the role of the output.
  - The other input of the cell becomes a conclusion.

Definition (Switching path)

Switching path = path that **does not contain** the two input arrows of a same $\parr$–cell.

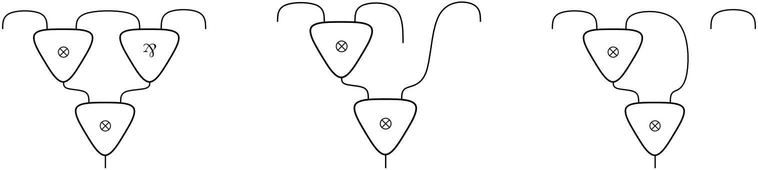
Danos–Regnier and Switch example

Theorem (Danos–Regnier)

Given an MLL proof structure S.

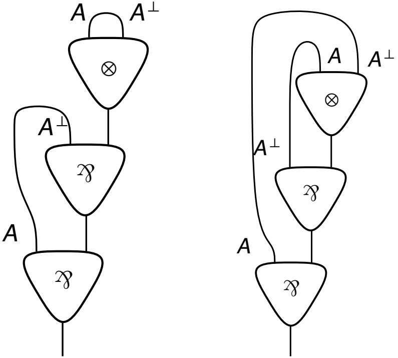
*S is a proof net $\Leftrightarrow$ All switch of S are **acyclic** and **connected**. $\Leftrightarrow$ No switching cycle + all switches connected .*

A proof structure, and its two switchings.



Proof–search = wiring

Consider proof–search on the formula $A \parr (A^{\perp}(A \otimes A^{\perp}))$.
First wiring does not correspond to a proof. Second wiring corresponds to a proof.

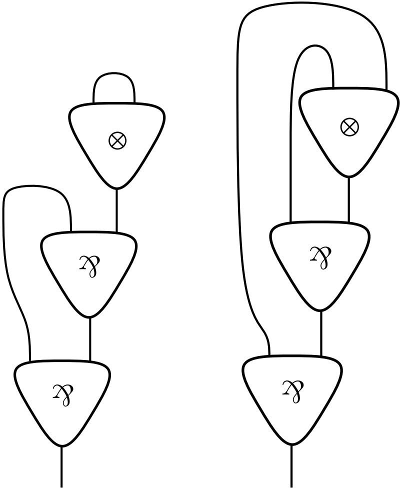


Untyped Proof structure.

Correctness is **purely geometrical**.

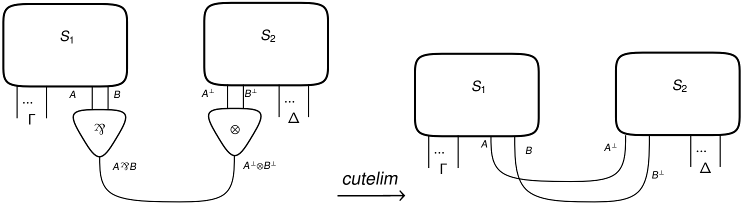
We can forget types, and reason on untyped structure that contain only the geometry of the proofs.

Proof = Correct Geometry + Wiring dual types.

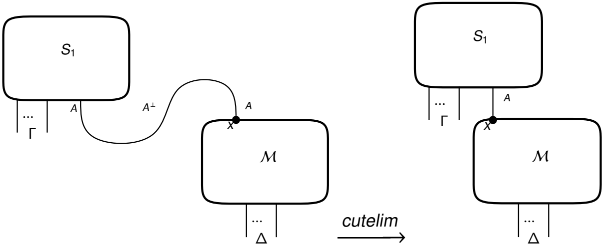


Cut elimination for proof structures

Multiplicative cut elimination:



Axiom cut elimination:

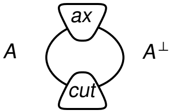


Bechet's Work

$$
\begin{tikzcd}
 & \text{Proof structures} \arrow[dl, bend right, "\text{its notion of specification}"'] \arrow[dr, bend left, "\text{its notion of calculus}"] & \\
\text{Correctness} \arrow[rr, bend right, leftrightarrow, red] & & \text{Cut–elimination}
\end{tikzcd}
$$

Possible Links? $\rightarrow$ Behavior

Bad Behavior of proof structures



- basically wrong proof structure = deadlock OR disconnection
- wrong = $S \xrightarrow{cutelim} S'$ AND S' basically wrong
- basically bad = S cutted with **proofnets** $P_1, ..., P_n$ is wrong.
- bad = $\phi(S)$ is basically bad.

Bechet's Theorem, 1998

Theorem (Bechet)

For a proof structure ***without cuts*** *: Incorrectness* $\Leftrightarrow$ *Bad Behavior*

Our focus : *Incorrectness* $\Rightarrow$ *Bad*

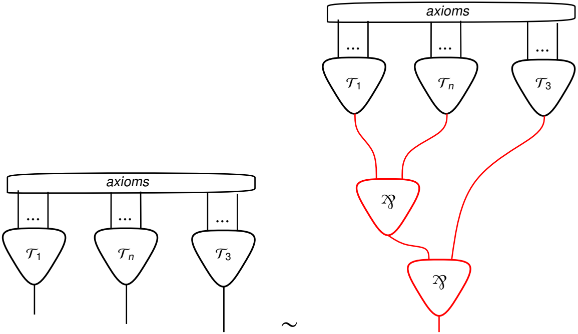
Two cases for switching : disconnected and cyclic

⅋–Closure and Behavior

Lemma

A proof structure has the same behavior than its ⅋–closure.

Here is an example of one ⅋–closure.



Dual of a Module.

Dual of a module = Negate types + switch ⅋ and $\otimes$

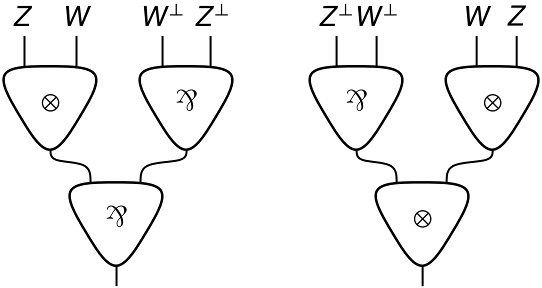
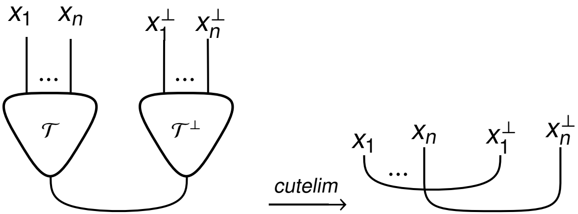


Figure 2: A module and its dual

Cut elimination with the Dual.

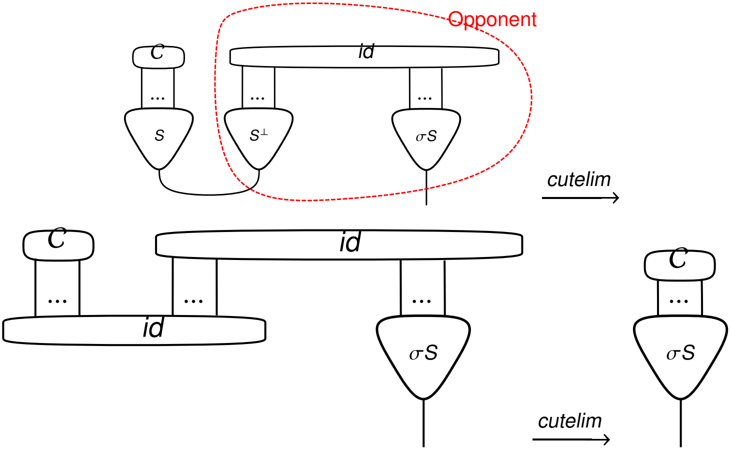
Theorem

The cut elimination of the interaction of tree $\mathcal{T}$ with its dual $\mathcal{T}^{\perp}$ reduces to a module made of only cuts, connecting the input arrows of each tree in a identity way (pair i with i).



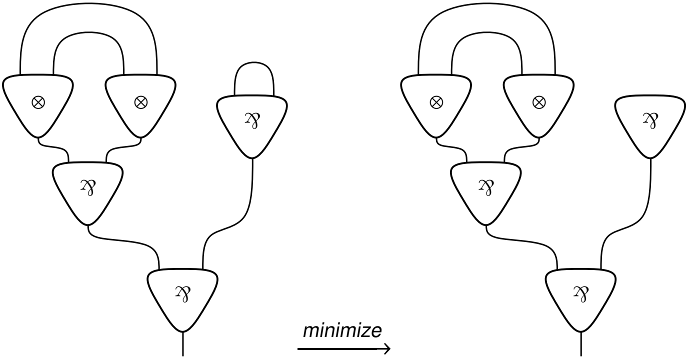
Opponent in disconnected switching case.

Assume S has a disconnected switch σS. S has a bad behavior:



Minimally cyclic Wiring

Configuration/Wiring will be denoted C.
Configuration = set of pairs of input arrows.
Cyclic configuration $\Rightarrow$ Minimally cyclic configuration.

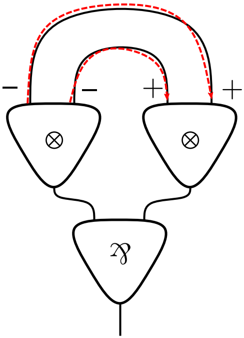


characterizing cycles

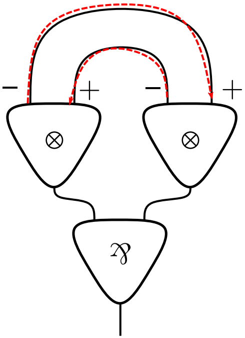
Cycles = sequence of polarized axiom meeting in $\otimes$.

(σ, p_1, p_2) where

- $\sigma : \mathcal{C} \to \mathcal{C}$ is a permutation with no invariant.
- $\pi_1, \pi_2 : \mathcal{C} \to$ *Arr* are choice functions that never coincide.



$(\alpha_1^-, \overline{\alpha}_1^+), (\alpha_2^-, \overline{\alpha}_2^+)$



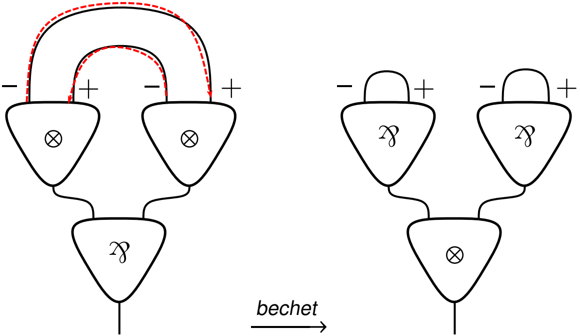
$(\alpha_1^-, \overline{\alpha}_1^+), (\alpha_2^+, \overline{\alpha}_2^-)$

Untyped Bechet Transformation

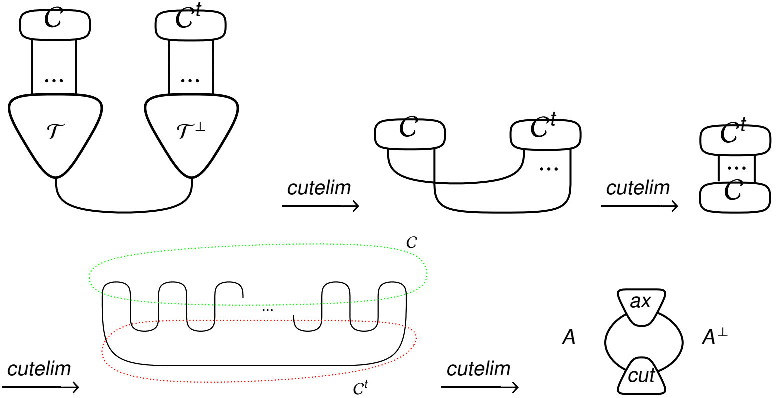
Bechet transformation : Minimal cyclic module → Acyclic module

$\mathcal{C} = \{p, \sigma(p), ..., \sigma^n(p)\}$

$\mathcal{C}^t = \{(\pi_2(p), \pi_1(\sigma p)) \mid p \in \mathcal{C}\}$



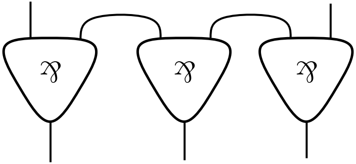
deadlock–reduction



Module correctness and Completion

Correct Module = Acyclic and No isolated component, For all switching.

Example of incorrect module

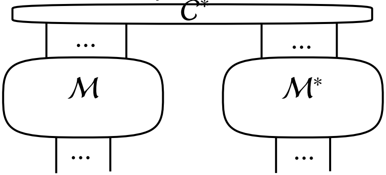


Module correctness and Completion

Theorem

Module Completion $\mathcal{M}$ Correct module $\Rightarrow$ $\mathcal{M}$ can be completed in a proof net.

We can complete $\mathcal{M}$ such that the following structure

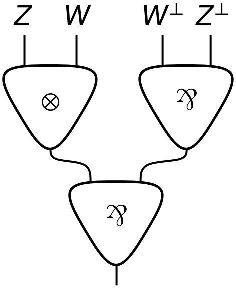


is a proof–net.

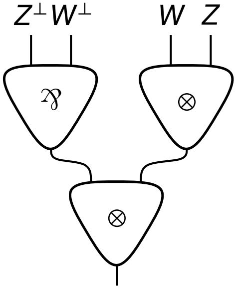
ϕ instantiation

Given any atomic variable X.

$\phi(X) =$



$\phi(X^\perp) =$

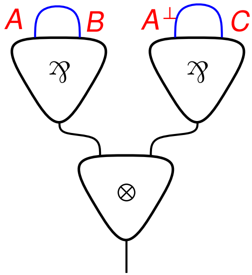


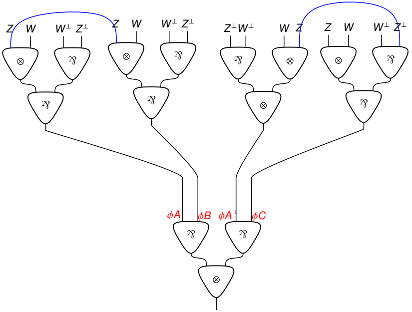
ϕ–extension and well–typedness

ϕ–extension S^{ϕ} = instantiation $\phi(S)$ + $\phi(X)$ tree–as–input + wiring $Z - Z^{\perp}$.

ϕ–extension allow for well typedness and conserve acyclicity.

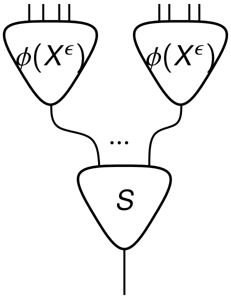
Below S and S^{ϕ}.





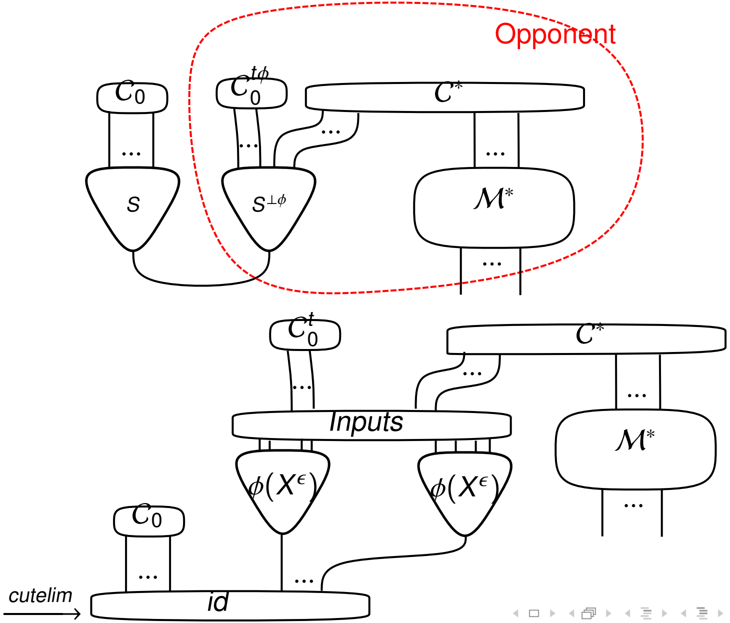
A ϕ–extension

Given some module S its ϕ–extension can be seen as the following.

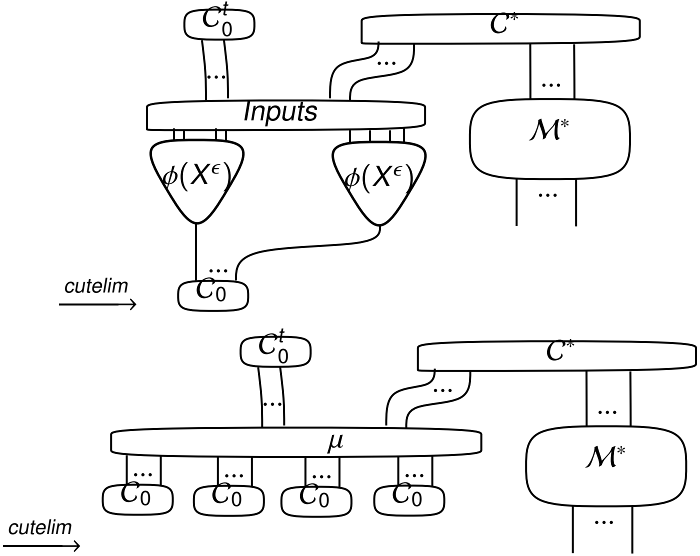
$S^{\phi} =$ 

Opponent in cyclic switching case. (1/3)

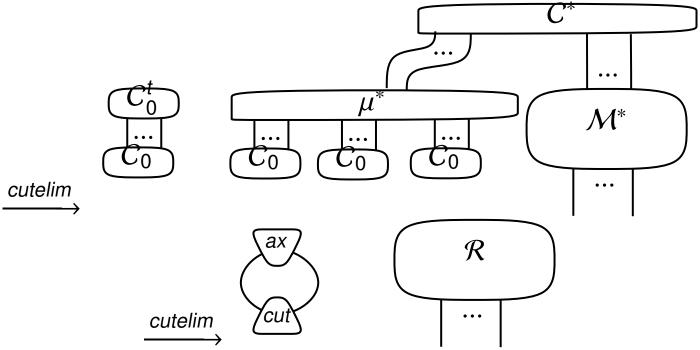
Given S a proof structure with a switching cycle.



Opponent in cyclic switching case. (2/3)



Opponent in cyclic switching case. (3/3)



Synthesis

We aim to show: *Incorrectness*$\Rightarrow$*Bad Behavior*.
We have two cases to treat for incorrectness:

- S has a disconnected switching.
- S has a cyclic switching.

In both cases we have created an opponent S^* such that:

- S^* is a proof net.
- For some instantiation ϕ, $\phi(S)$ interacting with S^* reduces to either a deadlock or a disconnected graph.

This allows to us to conclude (together with results of Danos–Regnier for the other direction)

Theorem (Bechet)

Given a proof structure S. S is incorrect $\Leftrightarrow$ S has a bad behavior.

Thanks

Thank you for following me. Feel free to ask any question.