APLL: A focusing-based automated prover for linear logic

Jui-Hsuan Wu

ENS Paris

December 18, 2018

Background

Some existing works :

- the focused inverse (forward) method by K. Chaudhuri
- llprover (http://bach.istc.kobe-u.ac.jp/llprover/) (without focusing, implemented in Prolog)
- another prover (https://gitlab.com/vcvpaiva/FLL-Prover) (backward method, implemented in Maude)

Overview of the prover

- two different logical systems (LL and ILL)
- two different proof methods (backward and forward)
- proof certificates using Coq and the Yalla library (only available for the backward method)
- LaTeX code of proofs (only available for the backward method)
- more details on https://github.com/wujuihsuan2016/LL_prover

Backward proof search

Our backward proof search consists of

- starting from the goal sequent,
- applying the rules backwards, and
- using the corresponding focused proof system

In **LLF**, there are two kinds of sequents : $\vdash \Theta : \Gamma \Uparrow L$ and $\vdash \Theta : \Gamma \Downarrow F$

Theorem 1 Proving $\vdash \Delta$ in LL is equivalent to proving $\vdash \cdot : \cdot \Uparrow \Delta$ in LLF.

Some rules:

$$\frac{}{\vdash \Theta : \Gamma \Uparrow L, \top}\ \top \qquad \frac{\vdash \Theta : \cdot \Uparrow F}{\vdash \Theta : \cdot \Downarrow\, !F}\ ! \qquad \frac{\vdash \Theta : \Gamma \Uparrow L, F \qquad \vdash \Theta : \Gamma \Uparrow L, G}{\vdash \Theta : \Gamma \Uparrow L, F \mathbin{\&} G}\ \&$$

$$\frac{\vdash \Theta : \Gamma \Downarrow F_1}{\vdash \Theta : \Gamma \Downarrow F_1 \oplus F_2}\ \oplus_1 \qquad \frac{\vdash \Theta : \Gamma \Downarrow F \qquad \vdash \Theta : \Gamma' \Downarrow G}{\vdash \Theta : \Gamma, \Gamma' \Downarrow F \otimes G}\ \otimes$$

$$\frac{\vdash \Theta : \Gamma \Downarrow F}{\vdash \Theta : \Gamma, F \Uparrow}\ D_1 \qquad \frac{\vdash \Theta, F : \Gamma \Downarrow F}{\vdash \Theta, F : \Gamma \Uparrow}\ D_2 \qquad (F \text{ is not a negative literal})$$

$$\frac{\vdash \Theta : \Gamma, S \Uparrow L}{\vdash \Theta : \Gamma \Uparrow L, S}\ R\Uparrow \qquad (S \text{ is not asynchronous})$$

$$\frac{\vdash \Theta : \Gamma \Uparrow N}{\vdash \Theta : \Gamma \Downarrow N}\ R\Downarrow \qquad (N \text{ is neither synchronous nor a positive atom})$$

Control of the D_2 rule

- The main factor that slows down the algorithm : the D_2 rule
- Problem : the classic recursive depth-first design will choose the same formula every time we apply the D_2 rule
- Solution : choose the formula in a round-robin style
- Implementation : a list of candidates select_d2 and an integer max_d2

Order of branches

- In rules such as & or $\otimes$, there are two branches to prove.
- whynot-height $wn_h(F)$ of a formula F :
  $wn_h(1) = wn_h(0) = wn_h(\top) = wn_h(\bot) = wn_h(X) = wn_h(X^\bot) = 0$ where X is an atom.

  $wn_h(?F) = 1 + wn_h(F), wn_h(!F) = wn_h(F), wn_h(F \diamond G) = max(wn_h(F), wn_h(G)), \quad \forall \diamond \in \{\otimes, \oplus, \&, \parr\}$
- By choosing the branch with smaller whynot-height, we are likely to visit the branch that can be proved (or disproved) faster than the other one.

Forward proof search: the focused inverse method

Idea : keep a database (set) of intermediate sequents and generate new provable sequents and add them into the database.
Main issue:

- How to use the stored sequents to generate new sequents?
- How to minimize the number of "useless" sequents?

Definition (forward sequents): A forward sequent has one of the following forms:

- $\Theta \; ; \; [\Gamma]_0$ (strong)
- $\Theta \; ; \; [\Gamma]_1$ (weak)

Intuitively, weak sequents are used to deal with arbitrary linear zones.

Ex: the conclusion of the $\top R$ of ILLF can be expressed by $\Theta \; ; \; [\cdot]_1$ and even by $\cdot \; ; \; [\cdot]_1$

Definition (subsumption or "weaker than"):
We define the subsumption relation between forward sequents as follows,

- $(\Theta\ ;\ [\Gamma]_0) \prec (\Theta'\ ;\ [\Gamma]_0)$ if $\Theta \subseteq \Theta'$
- $(\Theta\ ;\ [\Gamma]_1) \prec (\Theta'\ :\ [\Gamma']_w)$ if $\Theta \subseteq \Theta'$ and $\Gamma \subseteq \Gamma'$

Definition (Additive composition):
Given two linear contexts with weakness flag, $[\Gamma]_w$ and $[\Gamma']_{w'}$, we define their additive composition as follows:

$$[\Gamma]_w + [\Gamma']_{w'} = \begin{cases} [\Gamma]_0 & \text{if } w = w' = 0 \text{ and } \Gamma = \Gamma' \\ [\Gamma]_0 & \text{if } w = 0,\ w' = 1 \text{ and } \Gamma' \subseteq \Gamma \\ [\Gamma']_0 & \text{if } w = 1,\ w' = 0 \text{ and } \Gamma \subseteq \Gamma' \\ [\Gamma \sqcup \Gamma']_1 & \text{if } w = w' = 1 \end{cases}$$

where $\Gamma \sqcup \Gamma'$ denotes the least upper bound of Γ and Γ'.
Note that this function is partial.

Definition (Multiplicative composition):
Given two linear contexts with weakness flag, $[\Gamma]_w$ and $[\Gamma']_{w'}$, we define their multiplicative composition as follows:
$[\Gamma]_w \times [\Gamma']_{w'} = [\Gamma \cup \Gamma']_{w \vee w'}$
Here, $\Gamma \cup \Gamma'$ denotes the sum of the multisets Γ and Γ'.

Definition (Insertion):
Given a linear context with weakness flag $[\Gamma]_w$ and a proposition A, we define the result context after insertion of A as follows:
$[\Gamma]_w, A = [\Gamma, A]_w$

Derived rules of the forward calculus:

$$\frac{s_1 \;\; s_2 \cdots s_n \;\; (foc(P)[s_1 \cdot s_2 \cdots s_n] \hookrightarrow \Theta \; ; \; D)}{\Theta \; ; \; D, P} \; foc$$

$$\frac{s_1 \;\; s_2 \cdots s_n \;\; (foc(A)[s_1 \cdot s_2 \cdots s_n] \hookrightarrow \Theta \; ; \; D)}{\Theta, A \; ; \; D} \; ?foc$$

Focal phase:

$$\frac{foc(A)[\Sigma_1] \hookrightarrow \Theta_1 \ ; \ D_1 \qquad foc(B)[\Sigma_2] \hookrightarrow \Theta_2 \ ; \ D_2}{foc(A \otimes B)[\Sigma_1 \cdot \Sigma_2] \hookrightarrow \Theta_1, \Theta_2 \ ; \ D_1 \times D_2} \otimes F$$

$$\frac{foc(A_i)[\Sigma] \hookrightarrow s}{foc(A_1 \oplus A_2)[\Sigma] \hookrightarrow s} \oplus F_i \qquad \frac{act(\cdot \ ; \ \cdot \ ; A)[\Sigma] \hookrightarrow \Theta \ ; \ [\cdot]_w}{foc(!A)[\Sigma] \hookrightarrow \Theta \ ; \ [\cdot]_0} \ !F$$

$$\frac{}{foc(1)[\cdot] \hookrightarrow \cdot \ ; \ [\cdot]_0} \ 1F \qquad \frac{}{foc(X)[\cdot] \hookrightarrow \cdot \ ; \ [X^\perp]_0} \ init$$

$$\frac{act(\cdot \ ; \ \cdot \ ; L)[\Sigma] \hookrightarrow s}{foc(L)[\Sigma] \hookrightarrow s} \ FA$$ where L is asynchronous

Active phase:

$$\frac{act(\Theta \; ; \; \Gamma \; ; \; L, A)[\Sigma_1] \hookrightarrow \Theta_1 \; ; \; D_1 \qquad act(\Theta \; ; \; \Gamma \; ; \; L, B)[\Sigma_2] \hookrightarrow \Theta_2 \; ; \; D_2}{act(\Theta \; ; \; \Gamma \; ; \; L, A \,\&\, B)[\Sigma_1 \cdot \Sigma_2] \hookrightarrow \Theta_1, \Theta_2 \; ; \; D_1 + D_2} \,\&A$$

$$\frac{act(\Theta \; ; \; \Gamma \; ; \; L, A, B)[\Sigma] \hookrightarrow s}{act(\Theta \; ; \; \Gamma \; ; \; L, A \parr B)[\Sigma] \hookrightarrow s} \parr A \qquad \frac{}{act(\Theta \; ; \; \Gamma \; ; \; L, \top)[\Sigma] \hookrightarrow \cdot \; ; \; [\cdot]_1} \top A$$

$$\frac{act(\Theta \; ; \; \Gamma \; ; \; L, A_i)[\Sigma] \hookrightarrow \Theta' \; ; \; [\Gamma']_1}{act(\Theta \; ; \; \Gamma \; ; \; L, A_1 \parr A_2)[\Sigma] \hookrightarrow \Theta' \; ; \; [\Gamma']_1} \parr A_i$$

$$\frac{act(\Theta \cup \{A\} \; ; \; \Gamma \; ; \; L)[\Sigma] \hookrightarrow s}{act(\Theta \; ; \; \Gamma \; ; \; L, ?A)[\Sigma] \hookrightarrow s} \,?A$$

Active phase:

$$\frac{act(\Theta\ ;\ \Gamma\ ;\ L)[\Sigma] \hookrightarrow s}{act(\Theta\ ;\ \Gamma\ ;\ L, \bot)[\Sigma] \hookrightarrow s}\ \bot A$$

$$\frac{act(\Theta\ ;\ \Gamma, R\ ;\ L)[\Sigma] \hookrightarrow s}{act(\Theta\ ;\ \Gamma\ ;\ L, R)[\Sigma] \hookrightarrow s}\ act \quad \text{where } R \text{ is synchronous}$$

$$\frac{}{act(\Theta\ ;\ \Gamma\ ;\ \cdot)[\Theta, \Theta'\ ;\ \Gamma, \Gamma'] \hookrightarrow \Theta'\ ;\ \Gamma'}\ match$$

Theorem (completeness):
If $\vdash \Theta : \Gamma \Uparrow \cdot$ is provable, then either

- Θ' ; $[\Gamma]_0$ is provable for some $\Theta' \subseteq \Theta$, or
- Θ' ; $[\Gamma']_1$ is provable for some $\Theta' \subseteq \Theta$ and $\Gamma' \subseteq \Gamma$.

Theorem (soundness):
If Θ ; $[\Gamma]_0$ is provable, then $\vdash \Theta : \Gamma \Uparrow \cdot$ is provable.
If Θ ; $[\Gamma]_1$ is provable, then $\vdash \Theta : \Gamma' \Uparrow \cdot$ is provable for every $\Gamma' \supseteq \Gamma$.

The first issue in the implementation is to enumerate the propositions for which we need to derive inference rules. We first define two functions a (active) and f (focal) inductively:

$$f(X) = X \qquad f(X^-) = \emptyset \qquad f(!A) = a(A)$$
$$f(?A) = a(?A)$$
$$f(A \otimes B) = f(A \oplus B) = f(A) \bigcup f(B)$$
$$f(A \mathbin{\&} B) = f(A \parr B) = a(A \mathbin{\&} B) = a(A \parr B)$$
$$f(1) = f(0) = f(\top) = f(\bot) = \emptyset$$
$$a(X) = \{X\} \quad a(X^-) = \{X^-\} \quad a(!A) = !A \bigcup f(A)$$
$$a(?A) = \{A_?\} \bigcup f(A)$$
$$a(A \otimes B) = \{A \otimes B\} \bigcup f(A \otimes B) \quad a(A \oplus B) = \{A \oplus B\} \bigcup f(A \oplus B)$$
$$a(A \mathbin{\&} B) = a(A \parr B) = a(A) \bigcup a(B)$$
$$a(1) = a(0) = a(\top) = a(\bot) = \emptyset$$

Definition (frontier): Given a goal sequent $\Theta \,;\, \Gamma \Uparrow \cdot$, its frontier contains:

- all (top-level) propositions in Θ and Γ
- f(A) for all A in Θ and Γ.

Theorem:
In any backward focused proof, all sequents of the form $\Theta \,;\, \Gamma \Uparrow \cdot$ consists only of frontier propositions of the goal sequent.

Our implementation is quite simple:
For every frontier proposition, we consider all possible (multi-)lists we can form by using the sequents in the database and apply one of the derived rules (foc or ?foc). Note that we should set a bound on the number of copies of the same sequent we can have in these (multi-)lists. Hence, the completeness is not preserved but the soundness is.

An implementation technique about the unrestricted context (the ?foc rule) : if the proposition $A_?$ considered occurs in the unrestricted zone of the goal sequent $\Theta_0 \; ; \Gamma_0$, then it need not to be added into the unrestricted zone in the conclusion.

In the intuitionistic case, there is a foc^+ rule of the following form:

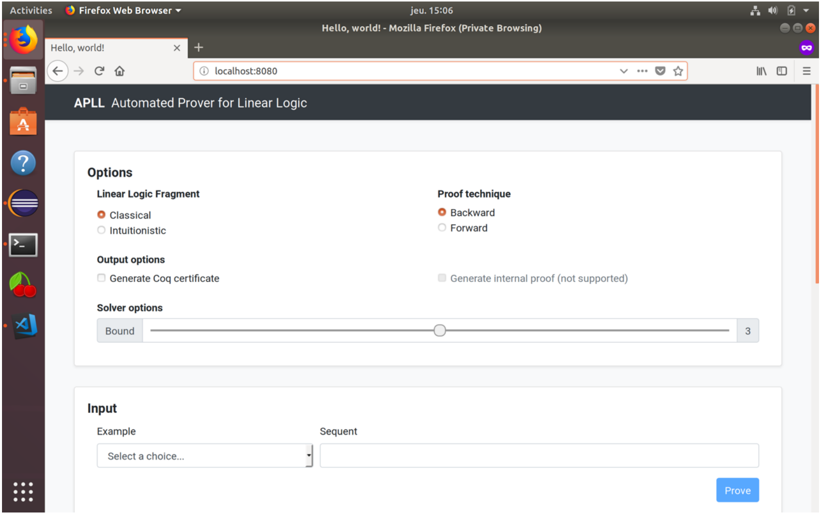
$$\frac{s_1 \;\; s_2 \cdots s_n \;\; (foc^+(Q)[s_1 \cdot s_2 \cdots s_n] \hookrightarrow \Theta \; ; \; D \longrightarrow \cdot)}{\Theta \; ; \; D \longrightarrow Q} \; foc^+$$

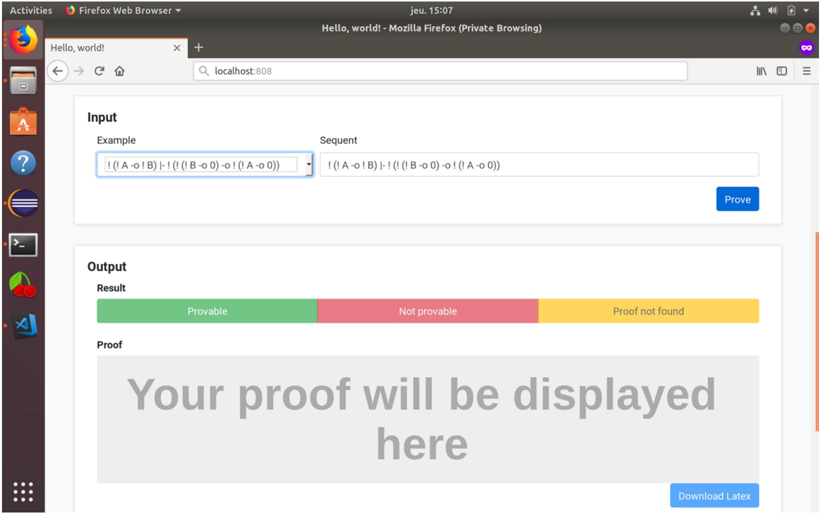
If every occurrence of Q in the goal sequent is of the form $!^k Q$, then we can directly add $!^k Q$ into the conclusion.

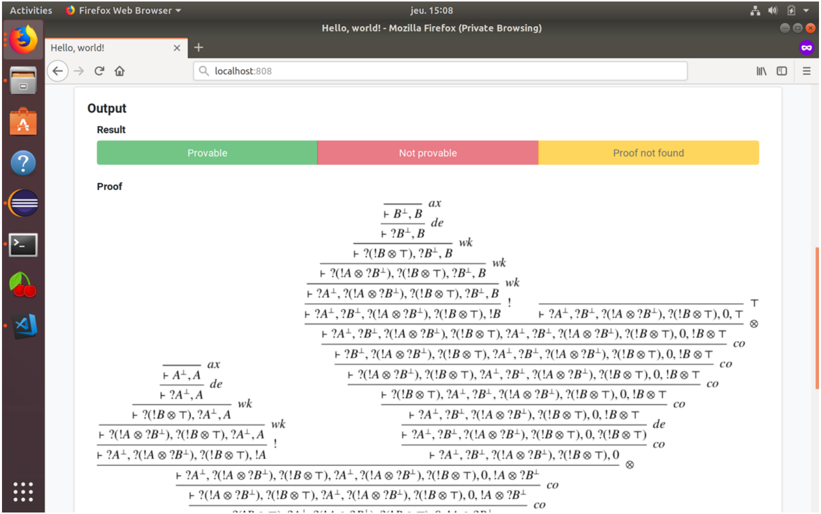
Tests and Results

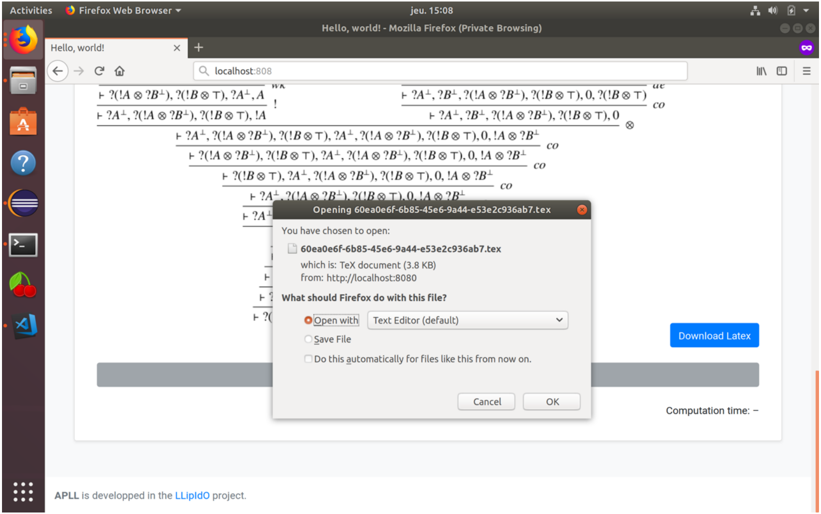
- Most of the tests are gathered from https://github.com/carlosolarte/Benchmarking-Linear-Logic/tree/master/TPTP which are obtained by translation from intuitionistic logic to linear logic.
- In general, the inverse (forward) method works faster than the backward method. In some cases, the former works worse because of the redundancy of the database of sequents.

A web interface for the prover under construction




Activities
Firefox Web Browser
jeu. 15:07
Hello, world! - Mozilla Firefox (Private Browsing)
Hello, world!
localhost:808
Input
Example
! (! A -o ! B) |- ! (! (! B -o 0) -o ! (! A -o 0))
Sequent
! (! A -o ! B) |- ! (! (! B -o 0) -o ! (! A -o 0))
Prove
Output
Result
Provable
Not provable
Proof not found
Proof
Your proof will be displayed here
Download Latex


Activities
Firefox Web Browser
jeu. 15:08
Hello, world! - Mozilla Firefox (Private Browsing)
Hello, world!
localhost:808
Output
Result
Provable
Not provable
Proof not found
Proof
$\vdash B^\perp, B$ ax
$\vdash ?B^\perp, B$ de
$\vdash ?(!B \otimes \top), ?B^\perp, B$ wk
$\vdash ?(!A \otimes ?B^\perp), ?(!B \otimes \top), ?B^\perp, B$ wk
$\vdash ?A^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), ?B^\perp, B$ wk
$\vdash ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), !B$!
$\vdash ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0, \top$ $\top$
$\vdash ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0, !B \otimes \top$ $\otimes$
$\vdash ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0, !B \otimes \top$ co
$\vdash ?(!A \otimes ?B^\perp), ?(!B \otimes \top), ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0, !B \otimes \top$ co
$\vdash ?(!B \otimes \top), ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0, !B \otimes \top$ co
$\vdash ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0, !B \otimes \top$ co
$\vdash ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0, ?(!B \otimes \top)$ de
$\vdash ?A^\perp, ?B^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0$ co
$\vdash A^\perp, A$ ax
$\vdash ?A^\perp, A$ de
$\vdash ?(!B \otimes \top), ?A^\perp, A$ wk
$\vdash ?(!A \otimes ?B^\perp), ?(!B \otimes \top), ?A^\perp, A$ wk
$\vdash ?A^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), !A$!
$\vdash ?A^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), ?A^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0, !A \otimes ?B^\perp$ $\otimes$
$\vdash ?(!A \otimes ?B^\perp), ?(!B \otimes \top), ?A^\perp, ?(!A \otimes ?B^\perp), ?(!B \otimes \top), 0, !A \otimes ?B^\perp$ co
co


Activities
Firefox Web Browser
jeu. 15:08
Hello, world! - Mozilla Firefox (Private Browsing)
Hello, world!
localhost:808
Opening 60ea0e6f-6b85-45e6-9a44-e53e2c936ab7.tex
You have chosen to open:
60ea0e6f-6b85-45e6-9a44-e53e2c936ab7.tex
which is: TeX document (3.8 KB)
from: http://localhost:8080
What should Firefox do with this file?
Open with
Text Editor (default)
Save File
Do this automatically for files like this from now on.
Cancel
OK
Download Latex
Computation time: –
APLL is developped in the LLipIdO project.

Demonstration

Future work

- Proof extraction (Coq export and Latex export) for the forward method.
- Extend benchmark tests and give more detailed analysis of the excution times.
- Define new Coq tactics in order to reduce the compile time.
- Use additional criteria to accelerate the unprovable cases.

References

1. J.-Y. Girard. Linear Logic. *Theoretical Computer Science*, 50:1-102, 1987.
2. J.-M. Andreoli. Logic programming with Focusing Proofs in Linear Logic. *Journal of Logic and Computation*, 2(3):297-347, 1992.
3. K. Chaudhuri. The focused inverse method for linear logic, Ph.D. Thesis, Carnegie Mellon University, Technical Report CMU-CS-06-162, 2006.
4. K. Chaudhuri and F. Pfenning. Focusing the inverse method for linear logic. In L. Ong, ed., *Proceedings of CSL 2005*, pages 200-215. Springer-Verlag LNCS 3634, 2005.
5. C. Liang and D. Miller. Focusing and polarization in linear, intuitionistic and classical logics. *Theoretical Computer Science*, 410(46):4747-4768, 2009.
6. C. Olarte, V. de Paiva, E. Pimentel and G. Reis. Benchmarking Linear Logic Translations. https://github.com/carlosolarte/Benchmarking-Linear-Logic