

Correctness of Multiplicative (and Exponential) Proof Structures is *NL*-Complete

Paulin Jacobé de Naurois¹ Virgile Mogbil¹

LIPN, Université Paris 13, France

Séminaire LDP, 29 Mai 2008

Motivations

- The **proof nets** of Linear logic (LL, Girard87):
- non-sequential graph-theoretic representation of proofs,
 - local and parallel cut-elimination,
 - sequentializable i.e correspond to sequent calculus derivations.
- The **proof structures** are freely built on the same syntax.

Main LL decision problems:

- deciding the provability of a given formula,
- deciding if two given proofs reduce to the same normal form,
- deciding the correctness of a given proof structure, i.e. whether it comes from a sequent calculus derivation. One uses a **correctness criterion** to distinguish proof nets among proof structures.

fragment		decision problem		
	units	provability	cut-elimination	correctness
MLL	no	<i>NP</i> -complete	<i>P</i> -complete	NL-complete
MELL	no/yes	open	(at most non-elem.)	

The long story of correctness criteria:

- **Long-trip** (Girard87) is based on travels and was the first one.
- **Acyclic-Connected** (DanosRegnier89) is a condition is based on switchings i.e. the choice of one premise for each \wp connective. The condition is that all the associated graphs are trees. A naive implementation of Acyclic-Connected uses exponential time.
- **Contractibility** (Danos90) is done in quadratic time by repeating two graph rewriting rules until one obtains a simple node.
- **Graph Parsing** (Lafont95) is a strategy for Contractibility which is implemented in linear time as a sort of unification (Guerrini99).
- **Dominator Tree** (MurawskiOng00/06) is a linear time correctness criterion for essential nets (IMLL), to which proof structures correctness reduces in linear time.
- **Ribbon** (Melliès04) is a topological condition requiring homeomorphism to the disk.

Outline

- 1 Multiplicative Linear Logic (MLL) and Correctness
- 2 Complexity classes and related problems
- 3 New correctness criterion
- 4 NL-Completeness of the new criterion

Multiplicative Linear Logic (MLL)

- Formulae:

$$F ::= A \mid A^\perp \mid F \otimes F \mid F \wp F$$

- Sequent:

$$\vdash \Gamma$$

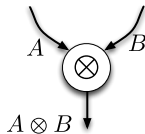
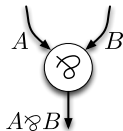
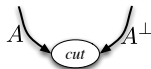
where Γ is a multiset of formulae, so that exchange is implicit.

- Sequent calculus: No weakening, no contraction.

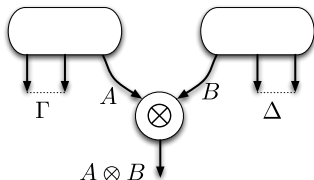
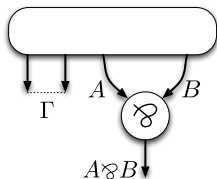
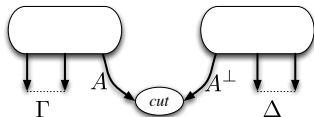
$$\frac{}{\vdash A, A^\perp} \text{ (ax)} \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} \text{ (cut)}$$

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

- Proof structures:



- **Proof nets**: inferred from sequent calculus rules.

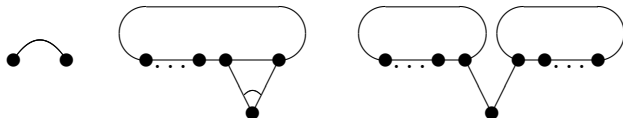


Paired graph

Definition

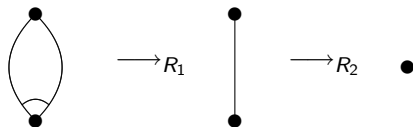
A *paired graph* is an undirected graph $G = (V, E)$ with a set of *pairs* $C(G) \subseteq E \times E$ which are pairwise disjoint couples of edges with the same target, called a *pair-node*, and two (possibly distinct) sources called the *premise-nodes*.

Paired graph constructors associated to MLL proof nets: respectively *ax-link*, \wp -link and \otimes -link



Contraction correctness criterion

Contraction rules \rightarrow_c : paired graphs rewriting rules on distinct nodes.



Notations:

- $G[\forall \mapsto \setminus \cdot]$ is the subgraph of G where the (abusively speaking) left edge of each pair of $C(G)$ is erased.
- $G \rightarrow_c^* \bullet$ denotes that G contracts to a single vertex with no edge.

Theorem (Correctness criterion (Danos90))

A MLL proof structure R is a MLL proof net iff $G_R \rightarrow_c^* \bullet$.

Some complexity classes

The following inclusion results are also well known:

$$AC^0 \subseteq L \subseteq NL \subseteq AC^1 \subseteq P$$

where it remains unknown whether any of these inclusions is strict.

- AC^0 (respectively AC^1) is the class of problems solvable by a uniform family of circuits of constant (resp. logarithmic) depth and polynomial size, with NOT gates and AND, OR gates of unbounded fan-in.
- L is the class of problems solvable by a deterministic Turing machine which only uses a logarithmic working space.
- $coNL$ (respectively NL) is the class of problems solvable by a NTM which only uses a logarithmic working space, such that:
 - If the answer is "no", at least one (resp. all) computation path rejects.
 - If the answer is "yes", all (resp. at least one) computation paths accept.

Theorem (Immerman87, Szelepcsényi87)

$$NL = coNL.$$

Related problems

SOURCE-TARGET CONNECTIVITY (STCONN):

Given a directed graph $G = (V, E)$ and two vertices s and t , is there a path from s to t in G ?

STCONN is NL -complete under constant-depth reductions (JonesAI.76).

UNDIRECTED SOURCE-TARGET CONNECTIVITY (USTCONN):

Given an undirected graph $G = (V, E)$ and two vertices s and t , do s and t belong to the same connected component of G ?

USTCONN is L -complete under constant-depth reductions (Reingold05).

MLL CORRECTNESS (MLL-CORR):

Given a MLL proof structure R , is it a MLL proof net?

A related problem: SDAG

UNIVERSAL SOURCE DAG (SDAG):

Given a directed graph $G = (V, E)$, is it acyclic and does there exist a source node s such that there is a path from s to each vertex ?

Theorem

SDAG is *NL-complete under constant-depth reductions*

Proof of the *NL*-membership.

- acyclicity:

$$\forall (x, y) \in V^2, \neg \text{STCONN}(G, x, y) \vee \neg \text{STCONN}(G, y, x).$$

- universal source: $\forall s \in V (\forall x \in V, \text{STCONN}(G, s, x))$



Proof sketch

Lemma (hardness)

SDAG is *coNL-hard* under constant-depth reductions.

i) $\mathcal{L} \in \text{coNL}$: decided by a NTM M in space $\leq k \log(n)$ on inputs of size n
 \mathcal{C}_n : set of configurations of M of size $\leq k \log(n)$.

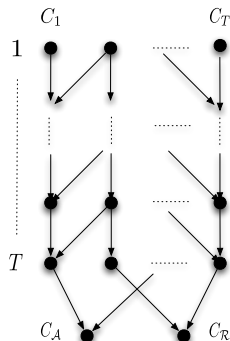
$T = |\mathcal{C}_n| \leq n^k$: upper bound for the computation time on inputs of size n .
Configuration = either accepting or rejecting.

Proof sketch

Lemma (hardness)

SDAG is coNL-hard under constant-depth reductions.

ii) Let us consider the following directed graph:
(Temporized conguration graph)

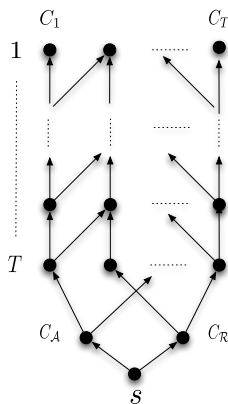


Proof sketch

Lemma (hardness)

SDAG is coNL-hard under constant-depth reductions.

- ii) Let us consider the following directed graph:
(Temporized conuguration graph)
(Path = decreasing sequence of t_i , $\wedge s = \text{source}$)
 $\Rightarrow G_n$ satisfies SDAG.



Proof sketch

Lemma (hardness)

SDAG is coNL-hard under constant-depth reductions.

ii) Let us consider the following directed graph:
(Temporized conguration graph)

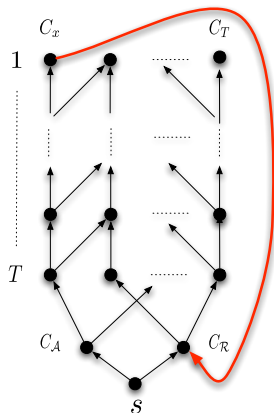
(Path = decreasing sequence of t_i , $\wedge s = \text{source}$)

$\Rightarrow G_n$ satisfies SDAG.

(Let $G_n^x = G_n \cup \{(C_x, 1) \rightarrow C_{\mathcal{R}}\}$)

$\Rightarrow G_n^x$ satisfies SDAG iff $x \in \mathcal{L}$.

Configuration graph = computed with constant-depth circuit \Rightarrow idem G_n^x .



$x \in \mathcal{L}$

Proof sketch

Lemma (hardness)

SDAG is coNL-hard under constant-depth reductions.

ii) Let us consider the following directed graph:
(Temporized conguration graph)

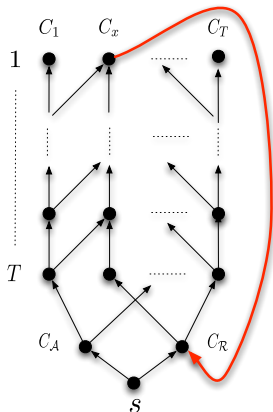
(Path = decreasing sequence of t_i , \wedge s = source)

$\Rightarrow G_n$ satisfies SDAG.

(Let $G_n^x = G_n \cup \{(C_x, 1) \rightarrow C_R\}$)

$\Rightarrow G_n^x$ satisfies SDAG iff $x \in \mathcal{L}$.

Configuration graph = computed with constant-depth circuit \Rightarrow idem G_n^x .



$x \notin \mathcal{L}$

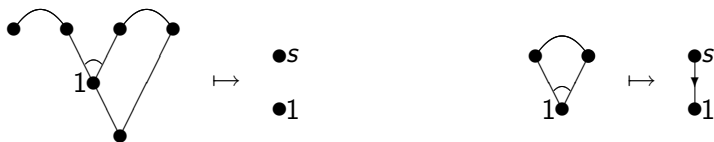
New correctness criterion (1)

For a given proof net, the following notion of dependency graph provides a partial order among its \wp -nodes corresponding to some valid contraction sequences accordingly to rule R_1 .

Definition (dependency graph $D(G)$ of a paired graph G)

Directed graph whose vertices are the pair-nodes of G and a new vertex s . Edges are defined to a node x when there exists an elementary path $p_x = x_1, \dots, x_r$ in $G[\forall \mapsto \lambda \cdot]$, $x \notin p_x$, and:

- $(s \rightarrow x)$ if $\forall y$ pair-node, $y \notin p_x$,
- $(y \rightarrow x)$ if $y \in$ every elementary path $p_x = x_1, \dots, x_r$ in $G[\forall \mapsto \lambda \cdot]$.



New correctness criterion (2)

Theorem (Correctness Criterion)

A MLL proof structure R is a MLL proof net if and only if:

- $D(G_R)$ satisfies SDAG, and
- $G_R[\forall \mapsto \lambda \cdot]$ is a tree.

Lemma

If $G \rightarrow_c^* \bullet$ then $D(G)$ satisfies SDAG.

Proof by construction of $D(G)$ because $G[\forall \mapsto \lambda \cdot]$ is a tree (when $G \rightarrow_c^* \bullet$).

Lemma

Let G be a paired graph such that $G[\forall \mapsto \lambda \cdot]$ is a tree. If the dependency graph $D(G)$ of G satisfies SDAG then $G \rightarrow_c^* \bullet$.

Proof by induction on the length of the paths from the source: one built a sequence of contracted nodes of G .

NL-Completeness of the new criterion

Theorem

MLL-CORR is NL-complete under constant-depth reductions.

Lemma (membership)

MLL-CORR \in NL.

Proof:

- i) the functions $R \mapsto G_R$ and $G_R \mapsto D(G_R)$ are computable in *FL*.
- ii) Checking that $G_R[\forall \mapsto \cdot]$ is a tree is doable in *L*.
- iii) Checking that $D(G_R)$ satisfies SDAG can be done in *NL*, by composing the functions in *FL* with an *NL* algorithm for SDAG.

NL-Completeness of the new criterion

Theorem

MLL-CORR is NL-complete under constant-depth reductions.

Lemma (membership)

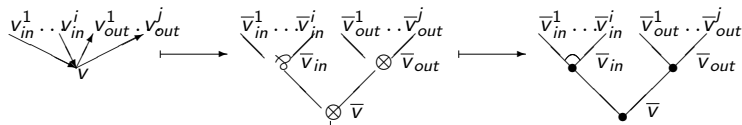
MLL-CORR \in NL.

Lemma (hardness)

MLL-CORR is NL-hard under constant-depth reductions.

Proof by reducing SDAG to MLL-CORR in two steps:

i) G directed graph $\mapsto S_G$ a proof structure:



ii) S_G is correct iff G_{S_G} its associated paired graph satisfies SDAG.

Conclusion

MLL-CORR is NL -complete under constant-depth reductions and exactly the same method applies for:

- Multiplicative Exponential LL (MELL without units),
- ("Degenerated") MELL with units,
- Intuitionistic MLL (IMLL). Remark that Murawski and Ong give an NL algorithm solving IMLL-CORR but MLL-CORR reduction is linear time and linear space.

Conclusion

MALL-CORR (MALL à la Hugues and Van Glabbeek) needs more than dependency graphs (see LICS'08).

fragment		decision problem		
	units	provability	cut-elimination	correctness
MLL	no	<i>NP</i> -complete	<i>P</i> -complete	NL-complete
MELL	yes	open	non-elementary	
MALL	no	<i>PSPACE</i> -complete	<i>coNP</i> -complete	