Notes on Light Linear Logic

Damiano Mazza
Institut de Mathématiques de Luminy
mazza@iml.univ-mrs.fr

November 13, 2003

1 Introduction

ELL and **LLL**, both introduced in [Gir98], are two logical systems which characterize respectively the class of Kalmar elementary functions and the class of deterministic polytime functions. Their syntactical formulation (both in the sequent calculus and proof-net framework) is quite different from that of "traditional" linear logic, namely because of the presence of *blocks* and *discharged formulas*. Even though the intended meaning of these two syntactical devices is well explained in Girard's paper (a block of formulas should be read as their additive disjunction, whereas a discharged formula $[A]$ should be read as $?A$), there is no obvious way to recover **ELL** and **LLL** proofs from the traditional syntax.

In [DJ99], Danos and Joinet addressed this issue in the case of **ELL**, redefining it as the proper sub-set of **LL** proofs which satisfy the so-called *stratification condition*. The stratification condition uses the concept of *exponential branch*, which can be recovered from the more general concept of *exponential tree*:

Definition 1 (Exponential trees and endoexponential nodes) *Let π be an* **LL** *proof-net. An* exponential tree *is a connected acyclic sub-graph τ of π whose edges are all labeled by the same formula $?A$. If τ's leaves are all axioms, derelictions and weakenings, then τ will be called* upward-maximal*; similarly, if τ's root is a node having $?A$ among its premises and $B \neq ?A$ as its conclusion (or having no conclusion at all), then τ will be called* downward-maximal. *If τ is both upward- and downward-maximal, then it will simply be called* maximal. *Moreover, we will refer to the root of a downward-maximal exponential tree as an* endoexponential node *(Danos and Joinet's definition doesn't mention endoexponential nodes, but we define them here because we will need them later on).*

Definition 2 (Exponential branch) *Let π be an* **LL** *proof-net. An* exponential branch *of π is a branch (from root to leaf) of a maximal exponential tree of π. The leaf-node will be called the* source *of the exponential branch.*

Danos and Joinet's system is defined as follows:

Definition 3 (ELL) **ELL** *is the sub-system of* **LL** *composed of all the proof-nets π which satisfy the* stratification condition*: any exponential branch of π*

whose source is a dereliction node (resp. an axiom node) crosses exactly one (resp. zero) exponential box(es).

The following result is proved in [DJ99]:

Theorem 1 (Danos-Joinet) ELL *is sound and complete with respect to the class of Kalmar elementary functions.*

In the next section we will investigate the possibility of extending this approach to **LLL**, and we will see that it is possible to separate by purely structural means (similar to Danos and Joinet's stratification condition) sub-systems of **LL** characterizing polytime; in particular, in section 3 we will define a system which corresponds exactly to Girard's **LLL**, as we will show in section 4. In section 5 we will prove the soundness and completeness of this latter system with respect to the class of deterministic polytime functions.

2 Polytime sub-systems of linear logic

We shall start by taking a look at the affine variants of **LLL** and **ELL**, i.e. **LAL** and **EAL**. The latter is simply obtained from full affine logic by "collapsing" promotion and dereliction into what is sometimes called *multifunctorial promotion*[1]. This operation corresponds to Danos and Joinet's stratification condition; as a matter of fact, in the absence of additive connectives, replacing promotion and dereliction by multifunctorial promotion in **LL** and imposing the stratification condition are perfectly equivalent.

As Girard himself states, "the main idea behind **LLL** is to restrict promotion to a unary context", and in fact **LAL** is obtained from **EAL** by allowing multifunctorial promotion only if the context contains at most one formula. But this idea cannot work in **LL**, because we can "cheat": a proof ending with

$$\dfrac{\dfrac{\dfrac{\vdash C, C, A}{\vdash ?C, ?C, A}}{\vdash ?C, A}}{\vdash ?C, !A}$$

respects both the stratification condition and the restriction to a unary context (of course assuming the sub-proof of $\vdash C, C, A$ does), but it is clearly able to duplicate a !-box and, what is worse, it actually uses both of its copies.

The solution comes from considering what we call *pax trees*:

Definition 4 (Pax tree) *Let π be an* **LL** *proof-net, and p an auxiliary gate of an exponential box of π. The* pax tree *of p is the upward-maximal exponential tree rooted at p.*

We must now add the paragraph modality to **LL**, since the lack of dereliction and the unary context restriction weaken too much the expressive power of the of course modality:

[1] Lafont calls it *soft promotion*.

Definition 5 (Obscene Linear Logic) *We add to the usual syntax of* **LL** *the self-dual modality* § *(*neutral*, or* paragraph*), with the following sequent calculus rule*[2]*:*

$$\frac{\vdash ?\Gamma, \Delta}{\vdash ?\Gamma, \S\Delta}$$

In the proof-net framework, this rule is translated by a §*-box, which may contain several auxiliary gates just like the usual* !*-box, each having its own pax tree. We call this "enriched" system* obscene linear logic[3]*,* **OLL***.*

Since **OLL** proof-nets may contain both !- and §-boxes, when speaking of an unspecified "exponential box" we will mean any of the two.

We can now define our first polytime sub-system of (obscene) linear logic:

Definition 6 ($\overline{\mathbf{LLL}}$) $\overline{\mathbf{LLL}}$ *is the sub-system of* **OLL** *composed of all the proof-nets* π *which satisfy the following conditions:*

- **i**. *Each* !*-box of* π *has at most one auxiliary gate.*
- **ii**. *Any exponential branch of* π *whose source is a dereliction node (resp. an axiom node) crosses exactly one (resp. zero) exponential box(es).*
- **iii**. *The pax tree of each* !*-box of* π *contains at most one dereliction leaf.*

Condition **ii** *is simply Danos and Joinet's condition, which here we call* downward stratification condition*;* **iii** *is called* upward stratification condition.

We can prove the following result:

Theorem 2 $\overline{\mathbf{LLL}}$ *is sound and complete with respect to the class of deterministic polytime functions.*

Proof. Both the soundness and completeness proof are more or less standard, and are achieved respectively by considering Girard's usual "round-by-round" reduction strategy and by encoding polytime Turing machines inside $\overline{\mathbf{LLL}}$. Details can be found in [Maz02]. □

The cut-elimination steps of $\overline{\mathbf{LLL}}$ are defined over a certain class of proof-nets, which respect the so-called *nouvelle syntaxe*. Every time we want to normalize a cut in a proof-net π and want make sure that the reduct stays inside $\overline{\mathbf{LLL}}$, we first apply certain rewriting rules in order to convert π to its nouvelle syntaxe representative, and then we perform the appropriate cut-elimination step. While this certainly leads to another $\overline{\mathbf{LLL}}$ proof-net, it can generate one that does not meet the nouvelle syntaxe criterion, so that if we want to reduce another cut, we have to apply the conversion again, and so on.

There is one remarkable point about nouvelle syntaxe: if we consider only proof-nets modulo the nouvelle syntaxe conversion, then condition **iii** becomes unnecessary. In fact, we can show the following:

Proposition 3 *Apart from its root and leaves, a pax tree of a nouvelle syntaxe proof-net only contains additive contraction nodes.*

[2]Notice that Δ may be empty; the necessity of this apparently useless "do-nothing" version of the rule is a consequence of the self-duality of §, and becomes clear when considering cut-elimination.

[3]This name was suggested by Harry Mairson, for reasons that would be too "off-topic" to be discussed here...

This assures us that even in the case where every leaf of the pax tree of the auxiliary gate of a !-box is a dereliction node, as normalization proceeds all of them except one will be sliced out, so that no duplication will occur. This is why condition **iii** is not needed anymore to obtain soundness.

In the general case, i.e. if we go back and consider also those proof-nets which are not nouvelle syntaxe, these considerations lead us to a reformulation of $\overline{\mathbf{LLL}}$, in which we weaken condition **iii** in the following way:

iii'. If c is a contraction node contained in the pax tree of the auxiliary gate of a !-box, then there can be dereliction nodes only in one of its two sub-trees.

Of course the system obtained by replacing **iii** with **iii**', which we refer to as *weak* $\overline{\mathbf{LLL}}$, is still sound and complete with respect to the polytime functions.

Actually, both $\overline{\mathbf{LLL}}$ and weak $\overline{\mathbf{LLL}}$ have two versions, according to how much liberal we are when fixing condition **i**. The version introduced in definition 6, which is the one considered in [Maz02], is what we call *liberal*; a *strict* version is obtained by restating condition **i** in order to allow only !-boxes having *exactly* one auxiliary gate.

The problem with all these systems is that they do not exactly correspond to the original **LLL**, as defined in [Gir98]. As a matter of fact, they all fail in proving one side of the fundamental "exponential isomorphism":

$$!A \otimes !B \multimap !(A \& B)$$

To prove it one has to necessarily make use of a promotion with a two-formula context, always forbidden by condition **i** both in its strict and liberal version.

Why is this a problem? After all, if all we want is a **PTIME** sound and complete sub-system of **LL**, $\overline{\mathbf{LLL}}$ and its variants perfectly do the job. But we strongly believe that logical systems corresponding to complexity classes should provide something more than yet another *syntactical characterization*; they should also be the portals to begin a *semantical study* of computational complexity, a possibility which is definitively open by the existence of a wide variety of denotational semantics for traditional linear logic. In particular, in [Bai02] Patrick Baillot has recently introduced the first candidates to such a role, i.e. *stratified coherent spaces*. Being based on the nowadays classical notion of coherence, these objects naturally satisfy the denotational isomorphism which $\overline{\mathbf{LLL}}$ rejects; as a consequence, it would be hard to adapt them to this system, just like it is seemingly difficult to transport stratified coherence onto the other polynomial systems like **LAL** and **SLL**, which both fail to include the exponential isomorphism.

Therefore, at present **LLL** seems to be the only complexity-bounded system with a denotational semantics, and this is why we think it is worth studying it in order to understand its workings as much as possible.

3 Recovering Girard's LLL

In the last section we drew inspiration from affine light systems and found various more or less equivalent subsystems of **OLL** characterizing polynomial time complexity, none of which though truly corresponds to **LLL**. To "patch" this failure, we shall start here by a thorough analysis of their main problem, namely the lack of the exponential isomorphism. The best thing is to look at how **LLL** proves it:

$$\dfrac{\dfrac{\dfrac{}{\vdash A^\perp; A}}{\vdash A^\perp, B^\perp; A} \qquad \dfrac{\dfrac{}{\vdash B^\perp; B}}{\vdash A^\perp, B^\perp; B}}{\dfrac{\vdash A^\perp, B^\perp; A\&B}{\dfrac{\vdash [A^\perp]; [B^\perp]; !(A\&B)}{\dfrac{\vdash ?A^\perp; ?B^\perp; !(A\&B)}{\vdash ?A^\perp \parr ?B^\perp; !(A\&B)}}}}$$

The pivot of the proof is the *additive weakening* rule, which lets us perform unrestricted weakening *provided we stay inside a block*. The promotion rule then takes *exactly one block*, treating it as a single formula, and "legalizes" the formulas inside the block by exponentiating them, i.e adding why nots in front of them.

But what are these devilish blocks anyway? By taking a quick look at **LLL**'s rules, we notice that blocks can be built only by means of additive weakening. It is possible to show that, in the absence of additives, this rule is useless. In fact, in multiplicative **LLL** a block is always made by exactly one "real" formula and a number of "fake" formulas, i.e. introduced by additive weakening. In this case, when a promotion legalizes a block it is really just transforming additive weakenings into multiplicative ones, so that the former may be avoided and replaced by the latter after the promotion, producing a perfectly equivalent proof.

If we introduce the additives, this is no longer true. In fact, a block can be "overlapped" with an identical one in the context of a &-rule, which is precisely what happens in the proof of the exponential isomorphism. In this situation, the resulting block may contain more than one "real" formula. The key idea to recover Girard's **LLL** from **OLL** is to look at this overlapping as a "quantum superimposition": in a &-rule, a "real" A on one side is matched against a "fake" A on the other side, making a "half-real" A.

We shall now formalize this intuition. First of all, let $\mathbb{Q}^+$ be the set of non-negative rationals, and $\tilde{\mathbb{Q}}^+ = \mathbb{Q}^+ \cup \{\infty\}$. The element ∞ behaves as follows:

$$\begin{array}{l} \forall x \in \tilde{\mathbb{Q}}^+, \quad \infty + x = x + \infty = \infty \\ \forall x \in \tilde{\mathbb{Q}}^+, x \neq 0, \quad \infty \cdot x = x \cdot \infty = \infty \\ \infty \cdot 0 = 0 \cdot \infty = 1 \end{array}$$

Let now π be an **OLL** proof-net. To each edge u of π which is labeled by an occurrence of an exponential formula $?A$ we assign a weight $\mho(u) \in \tilde{\mathbb{Q}}^+$:

- If u is the conclusion of an axiom node, $\mho(u) = \infty$.
- If u is the conclusion of a weakening node, $\mho(u) = 0$.
- If u is the conclusion of a dereliction node, $\mho(u) = 1$.
- If u is the conclusion of a contraction node whose premises are v and w, $\mho(u) = \mho(v) + \mho(w)$.

- If u is the conclusion of an auxiliary gate whose premise is v, $\mho(u) = \infty \cdot \mho(v) - 1$.
- To each additive box $\mathcal{A}$ of π we assign a *left bias* $\alpha \in \tilde{\mathbb{Q}}^+$ such that $0 < \alpha < 1$, its *right bias* being automatically determined as $1-\alpha$. If u is the conclusion of an additive contraction node of a box $\mathcal{A}$ of bias α whose left and right premises are respectively v and w, $\mho(u) = \alpha\mho(v) + (1-\alpha)\mho(w)$.

We shall assume that for any other edge u of π $\mho(u)$ is undefined.

We also assign a weight to exponential and additive boxes of π. In order to do this, we define the *pax set* of an exponential box $\mathcal{B}$ of π, noted $\mathsf{PaxSet}(\mathcal{B})$, as the set of the edges of π which are premise of an auxiliary gate of $\mathcal{B}$. Similarly, we define the *left wing* (resp. *right wing*) of an additive box $\mathcal{A}$ of π, noted $\mathsf{LWng}(\mathcal{A})$ (resp. $\mathsf{RWng}(\mathcal{A})$), as the set of edges of π whose weight is defined and *finite*, and which are the left (resp. right) premise of an additive contraction of $\mathcal{A}$. We can now define our weights as follows:

- If $\mathcal{B}$ is an exponential of π, then

$$\mho^{\mathsf{exp}}(\mathcal{B}) = \sum_{v \in \mathsf{PaxSet}(\mathcal{B})} \mho(v)$$

- If $\mathcal{A}$ is an additive box of π, then we define its left weight as

$$\mho^{\&}_{l}(\mathcal{A}) = \sum_{v \in \mathsf{LWng}(\mathcal{B})} \mho(v)$$

  and its right weight as

$$\mho^{\&}_{r}(\mathcal{A}) = \sum_{v \in \mathsf{RWng}(\mathcal{B})} \mho(v)$$

At this point we can introduce the sub-system of **OLL** which we will prove to be exactly Girard's **LLL**:

Definition 7 (LLL*) LLL* *is the sub-system of* **OLL** *composed of all the proof-nets π which satisfy the following conditions:*

Additive context *If $\mathcal{A}$ is an additive box of π, then $\mho^{\&}_{l}(\mathcal{A}) = \mho^{\&}_{r}(\mathcal{A})$.*

Inhibition *If u is a conclusion of π whose weight is defined, or the premise of an endoexponential node (see definition 1), then $\mho(u)$ is either null or infinite.*

Promotion context *If $\mathcal{B}$ is a $!$-box (resp. a $\S$-box) of π, then $\mho^{\mathsf{exp}}(\mathcal{B}) = 1$ (resp. $\mho^{\mathsf{exp}}(\mathcal{B}) < \infty$).*

The term "inhibition" refers to dereliction, i.e. the principle $!A \multimap A$ is not provable because of this condition. The following leads us to finally understand what Girard's blocks really are:

Lemma 4 *Under the additive context condition, the $\&$-rule preserves the weight of wings.*

Proof. By the additive context condition, we know that, if $\mathcal{A}$ is an additive box, $\mho_l^{\&}(\mathcal{A}) = \mho_r^{\&}(\mathcal{A}) = w$. If $u \in \mathsf{LWng}(\mathcal{A})$ and $v \in \mathsf{RWng}(\mathcal{A})$ are the two premises of an additive contraction of $\mathcal{A}$, we name its conclusion $u.v$. Suppose now that α is the left bias of $\mathcal{A}$. Then by definition of left and right weight we have

$$\begin{aligned}\sum_{u\in\mathsf{LWng}(\mathcal{A})}\sum_{v\in\mathsf{RWng}(\mathcal{A})}\mho(u.v) &= \sum_{u\in\mathsf{LWng}(\mathcal{A})}\sum_{v\in\mathsf{RWng}(\mathcal{A})}\alpha\mho(u)+(1-\alpha)\mho(v) = \\ &= \alpha\mho_l^{\&}(\mathcal{A})+(1-\alpha)\mho_r^{\&}(\mathcal{A}) = w\end{aligned}$$

So the sum of the finite weights of the conclusions of an additive box is equal to the weight of any of its wings (which is precisely the sum of the finite weights of its left or right context). ☐

Corollary 5 *If π is an* **OLL** *proof-net satisfying the additive context condition, the sum of the finite weights of its conclusions is always a non-negative integer.*

Proof. Finite weights are introduced by weakenings and derelictions, which both set the weight to a non-negative integer (respectively null and unitary). Contractions preserve integer weights, auxiliary gates preserve null weights and "saturate" to infinity other values; additive contractions introduce rational weights, but by lemma 4 the overall finite weight of the conclusion of their box is preserved, and therefore it is an integer if it was an integer. ☐

Now, the promotion context condition corresponds in **LLL** to the fact that the context of a promotion must be exactly one block; so *an additive block is nothing but a set of finitely weighed occurrences of why not formulas whose total weight is 1.* The sum of the finite weights of the conclusions of a derivation, which we have shown is always an integer under the additive context condition, is precisely the number of additive blocks in the corresponding **LLL** derivation (see lemma 8 below for a formal proof of this).

This means that the restriction we usually apply to the width of promotion contexts in light systems like **LAL** or $\overline{\mathbf{LLL}}$ is far too strict; here we finally came to understand that it is not so much the *width* that must be limited to 1, but the *weight*...

One last remark: if we liberalize the promotion context condition to $\mho^{\mathsf{exp}}(\mathcal{B}) < \infty$ for !-boxes also, we obtain **ELL**. After what we've just said about weights, restricting promotion to a finitely weighing context seems quite natural, since not doing so would correspond to an **LLL** promotion rule with an infinite number of blocks in the context!

4 Provability in LLL*

We will now show that, with respect to provability, **LLL*** and **LLL** are really the same logical system. Since **LLL** sequents are a little bit more complicated than traditional sequents (which are the ones considered in **LLL***), we need to decide how the conclusion of an **LLL** proof should be read, i.e. we need to interpret blocks and discharged formulas. Girard explains very well the meaning of these syntactical devices, so our translation is quite straight-forward (and going the other way around is even more trivial):

Definition 8 (Translations LLL$\rightarrow$ LLL* and LLL$^*\rightarrow$ LLL) *If $\vdash \Gamma$ is an* **LLL** *sequent, then $\vdash \Gamma^\star$ is the* **OLL** *sequent obtained from $\vdash \Gamma$ as follows:*

- *Change all blocks $A_1, \ldots, A_n$ into $A_1 \oplus \ldots \oplus A_n$.*
- *Change all discharged formulas $[A]$ into $?A$.*
- *Replace semi-colons with commas.*

Conversely, if $\vdash \Gamma$ is an **LLL*** *sequent, then $\vdash \Gamma^{;}$ is the* **LLL** *sequent obtained from $\vdash \Gamma$ by simply replacing all commas with semi-colons.*

Let us now prove the completeness of **LLL*** with respect to **LLL**. This first result shows that **LLL*** has what $\overline{\textbf{LLL}}$ lacked:

Lemma 6 **LLL*** *accepts the principle $!A \otimes !B \multimap !(A\&B)$.*

Proof. We will actually show that in **LLL*** one can derive an n-ary version of the "critical side" of the exponential isomorphism:

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash B_n, B_n^\perp}}{\vdash ?B_n, B_n^\perp}}{\vdash ?B_1, \ldots, ?B_n, B_n^\perp} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash B_2, B_2^\perp}}{\vdash ?B_2, B_2^\perp}}{\vdash ?B_1, \ldots, ?B_n, B_2^\perp} \qquad \dfrac{\dfrac{\dfrac{}{\vdash B_1, B_1^\perp}}{\vdash ?B_1, B_1^\perp}}{\vdash ?B_1, \ldots, ?B_n, B_1^\perp}}{\vdash ?B_1, \ldots, ?B_n, B_2^\perp \& B_1^\perp}}{\begin{matrix}\vdots\\ \vdash ?B_1, \ldots, ?B_n, B_{n-1}^\perp \& \ldots \& B_1^\perp\end{matrix}}}{\dfrac{\vdash ?B_1, \ldots, ?B_n, B_n^\perp \& \ldots \& B_1^\perp}{\vdash ?B_1, \ldots, ?B_n, !(B_n^\perp \& \ldots \& B_1^\perp)}}
$$

We need to check that this proof satisfies the three conditions of definition 7. Consider, for $i \in \{1, \ldots, n\}$, the sub-proofs

$$
\dfrac{\dfrac{\dfrac{}{\vdash B_i, B_i^\perp}}{\vdash ?B_i, B_i^\perp}}{\vdash ?B_1, \ldots, ?B_n, B_i^\perp}
$$

Notice that, when translating this into proof-nets, if u_j is the conclusion labeled by the occurrence of $?B_j$, then

$$
\begin{aligned}
&\mho(u_i) = 1\\
&\mho(u_j) = 0, \forall j \neq i,
\end{aligned}
$$

so that the sum of the finite weights over all conclusions is exactly 1 for every i. So the additive context condition is satisfied and, by lemma 4, the weight of the !-box associated to the last rule is still exactly 1, so that the promotion context condition is also met. For what concerns inhibition, there is no endoexponential node in the proof, and all the conclusions labeled by some $?B_i$ have either null or infinite weight, since they all come from auxiliary gates whose premise has a finite weight. $\square$

Now the completeness theorem:

Theorem 7 *If* $\vdash \Gamma$ *is provable in* **LLL**, *then* $\vdash \Gamma^\star$ *is provable in* $\mathbf{LLL}^*$.

Proof. By induction on the last rule that proves $\vdash \Gamma$. The only interesting cases are the additive structural rules and the exponential rules.

For what concerns the first ones, weakening is trivial, while contraction is simulated as follows:

$$\dfrac{\vdash \Gamma^\star, A \oplus B \oplus B \qquad \dfrac{\dfrac{\dfrac{}{\vdash B, B^\perp}}{\vdash A \oplus B, B^\perp} \qquad \dfrac{}{\vdash B^\perp \& A^\perp, A \oplus B}}{\vdash B^\perp \& B^\perp \& A^\perp, A \oplus B}}{\vdash \Gamma^\star, A \oplus B}$$

The why not rule (from $\vdash \Gamma; [A]$ derive $\vdash \Gamma; ?A$) need not be simulated, since our translation already encapsulates it, whereas multiplicative weakening and contraction are obtained by means of standard weakening and contraction in $\mathbf{LLL}^*$. To simulate the bang and paragraph rules, we need lemma 6:

$$\dfrac{\dfrac{\dfrac{\vdash B_1 \oplus \ldots \oplus B_n, A}{\vdash ?(B_1 \oplus \ldots \oplus B_n), A}}{\vdash ?(B_1 \oplus \ldots \oplus B_n), !A} \qquad \vdash ?B_1, \ldots, ?B_n, !(B_n^\perp \& \ldots \& B_1^\perp)}{\vdash ?B_1, \ldots, ?B_n, !A}$$

The simulation of the paragraph rule is perfectly analogous, and left to the reader. □

Now we must check that the converse holds as well. In order to do this, we need a few conventions. First of all, since we will be considering **OLL** sequent calculus proofs, we will commit the abuse of saying that an occurrence of a formula $?A$ has weight $\mho(?A) = w$ if the edge u of the corresponding proof-net matching that occurrence has weight w. Moreover, if $?\Delta$ is a sequence of why not formulas, we will denote by $\mho(?\Delta)$ the sum of their weights.

We start with the following general result, which formalizes the intuitive meaning of additive blocks we discussed in the previous section:

Lemma 8 (Provability of additive blocks) *If* $\vdash \Gamma, ?\Delta_1, \ldots, ?\Delta_n$ *is a provable* **OLL** *sequent such that, for all* i, $\mho(?\Delta_i) = 1$, *then the sequent*

$$\vdash \Gamma^{;}; \Delta_1; \ldots; \Delta_n$$

is provable in **LLL**.

Proof. By induction on the last rule of the derivation. If the last rule is an axiom, $n = 0$ and so the result holds trivially. For what concerns the induction step, the only non-trivial cases are those of a last rule concerning one of the $?\Delta_i$'s.

Suppose this is a dereliction, in which case one of the $?\Delta_i$ consists of a single formula. The proof therefore ends with a dereliction on $\vdash \Gamma, ?\Delta_1, \ldots, ?\Delta_{n-1}, A$ yielding $\vdash \Gamma, ?\Delta_1, \ldots, ?\Delta_{n-1}, ?A$. By induction we have a proof in **LLL** of $\vdash \Gamma^{;}; \Delta_1; \ldots; \Delta_{n-1}; A$, which is just what we need.

If it is a contraction, our proof ends like

$$\frac{\vdash \Gamma, ?\Delta_1, \ldots, ?\Delta_{n-1}, ?\Delta, ?A, ?A}{\vdash \Gamma, ?\Delta_1, \ldots, ?\Delta_{n-1}, ?\Delta, ?A}$$

with $\mho(?\Delta, ?A, ?A) = 1$. So in **LLL** we have a proof of

$$\vdash \Gamma^{¡}; \Delta_1; \ldots; \Delta_{n-1}; \Delta, A, A$$

from which we obtain $\vdash \Gamma^{¡}; \Delta_1; \ldots; \Delta_{n-1}; \Delta, A$ by additive contraction. Analogously, weakening is simulated by means of additive weakening. □

Here's how we finally prove soundness:

Theorem 9 *If* $\vdash \Gamma$ *is provable in* $\mathbf{LLL}^*$*, then* $\vdash \Gamma^{¡}$ *is provable in* **LLL**.

Proof. Again by induction on the last rule of the proof, for which the only interesting cases are as usual the exponential rules.

Weakening is simulated by means of a multiplicative weakening immediately followed by a why not rule; for what concerns contraction, we have the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash \Gamma^{¡}; ?A; ?A \qquad \dfrac{\dfrac{}{\vdash A^{\perp}; A}}{\vdash !A^{\perp}; [A]}}{\vdash \Gamma^{¡}; [A]; ?A} \qquad \dfrac{\dfrac{}{\vdash A^{\perp}; A}}{\vdash !A^{\perp}; [A]}}{\vdash \Gamma^{¡}; [A]; [A]}}{\vdash \Gamma^{¡}; [A]}}{\vdash \Gamma^{¡}; ?A}$$

Dereliction needs not be considered, since an **OLL** proof ending with this rule would not satisfy the inhibition condition and would therefore not be in $\mathbf{LLL}^*$.

Lemma 8 is needed for promotion. In fact, in this case we cannot reason by induction, since a proof ending with a promotion yielding $\vdash ?B_1, \ldots, ?B_n, !A$ comes necessarily from $\vdash ?B_1, \ldots, ?B_n, A$, whose proof is *not* in $\mathbf{LLL}^*$, again because of inhibition (by the promotion context condition, $\mho(?B_1, \ldots, ?B_n) = 1$, which means that at least one of the $?B_i$ has a finite non-null weight). But the sub-proof of $\vdash ?B_1, \ldots, ?B_n, A$ is certainly an **OLL** derivation, so lemma 8 holds and we obtain a proof of $\vdash B_1, \ldots, B_n; A$ in **LLL**, from which it easy to obtain, by an of course rule and several why not rules, a proof of the sequent $\vdash ?B_1; \ldots; ?B_n; !A$.

The paragraph case is perfectly similar to the one just considered and is left to the reader. □

5 Polytime soundness and completeness

Since, from the point of view of provability, $\mathbf{LLL}^*$ is equivalent to **LLL**, **FP**-completeness holds trivially by translating Girard's proof[4] [Gir98] into our system.

[4] Which by the way was incorrect, as noticed by Roversi in [Rov99]. Nevertheless, Roversi's proof adapts to **LLL** as well, so we can refer to that one instead.

For what concerns soundness, we will consider the following algorithm for reducing $\mathbf{LLL}^*$ proof-nets, which is absolutely not nice but has the good taste of being polytime and, most important of all, it is very easy to prove that it is actually so:

- Let π be the $\mathbf{LLL}^*$ proof-net we want to normalize; then we first translate π into the corresponding **LLL** proof-net $\pi^\bullet$. Lemma 10 will show that this operation can be carried over in quadratic time (in the size of π) and yields proof-nets of size linear in the original size.
- Normalize $\pi^\bullet$ with Girard's algorithm, which we know very well has the bound s^{2^d}, where s is the size of $\pi^\bullet$ and d its depth. This gives us an **LLL** proof-net σ, which is cut-free in the lazy case.
- Reconvert the result σ into an $\mathbf{LLL}^*$ proof-net $\sigma_\bullet$, an operation which lemma 11 will prove is again polynomial in the size of σ.

If the lemmas we referred to actually hold, we have obviously defined a procedure which, taken a *lazy*[5] $\mathbf{LLL}^*$ proof-net, returns its normal form in a time which is polynomial in the size of the starting proof-net.

We first need to define the two translations $(\cdot)^\bullet$ and $(\cdot)_\bullet$. Let us start with the first one. Take any $\mathbf{LLL}^*$ proof-net π, of size s and exponential depth d. We begin by calculating the weight $\mho(u)$ of each why not edge of π; when doing this, if we find out that $\mho(u) < \infty$, we replace the $?A$ labeling u with $\flat A$. After this procedure, which is obviously linear in s, we end up with a net $\pi^{\mho_0}$, which is just π with some why not edges decorated in a slightly different way.

The next thing we do is change back the label of the conclusions of those weakening nodes which we call *infinitary*:

Definition 9 (Infinitary weakening node) *Let π be an* $\mathbf{LLL}^*$ *proof-net, and w a weakening node of π, which is a leaf of an exponential tree τ. If among τ's leaves there are no dereliction nodes, then w is said to be* infinitary.

Now let u be the conclusion of an infinitary weakening of $\pi^{\mho_0}$. If its conclusion is labeled by $\flat A$, we change it back to $?A$. After having done that, we also chase down the path that leads from u to a conclusion (or a cut) of $\pi^{\mho_0}$, restoring all occurrences of $\flat A$ into $?A$. This operation needs to be carried on for each infinitary weakening; but finding all of them and doing the renaming cannot take more than quadratic time[6].

What we are left with is a net $\pi^{\mho}$ whose size and depth are still s and d, on which we finally start our translation:

Definition 10 (Proof-net translation from $\mathbf{LLL}^*$ to LLL) *We translate $\pi^{\mho}$ to the* **LLL** *proof-net $\pi^\bullet$ in the following way:*

[5] Laziness is imposed by Girard's procedure; it is obviously inherited by the algorithm we have defined. By the way, even if we tried to define a normalization procedure of our own, laziness would still be strictly required, since $\mathbf{LLL}^*$ (and **LLL**) is *not* strongly polytime, like **LAL** (see [Ter01]).

[6] If we have a weakening, we can check in linear time if it is infinitary by simply extracting its exponential tree; renaming hereditarily its conclusions is also linear. Since we need to do this for *all* weakenings, the result is quadratic. Better algorithms can probably be devised, but we don't really care about optimizations here.

- *Identity links (axioms and cuts) and logical (multiplicative and additive) links are translated to their respective* **LLL** *counterpart. The only difference concerns additive boxes, which are not present in* **LLL** *but are replaced by boolean weights.*
- *Derelictions are erased.*
- *Infinitary weakenings are translated to 0-ary why not nodes; the other weakening nodes are handled by additive weakenings in* **LLL**.
- *A contraction of premises $?A$ is translated as in the proof of theorem 9; a contraction of premises $\flat A$ is converted by means of additive contractions (a contraction in which one premise is $\flat A$ and the other $?A$ is forbidden by inhibition and the promotion context condition).*
- $!$*-boxes and* $\S$*-boxes are translated to their counterparts, and a series of unary why not nodes are attached to their discharged conclusions.*

Now the announced result:

Lemma 10 *The procedure for transforming a proof-net π of* **LLL*** *into the corresponding proof-net $\pi^\bullet$ of* **LLL** *is such that:*

a. its runtime is polynomial in the size of π;

b. the size of $\pi^\bullet$ is linear in the size of π.

Proof. Part a is a consequence of what we've discussed up to this point; for part b, notice that the only two cases in which the size could increase are the translation of a contraction of two $?A$ occurrences and the translation of an exponential box. The first case clearly adds a constant to the size, thus keeping it linear on the overall.

In the case of an exponential box, a number of unary why not nodes is added, and this number is equal to the width of the context of the $!$ or $\S$ rule. This width cannot be reasonably bounded if not by the size itself, which would cause exponential blow-ups. Fortunately the following argument straightens things out.

Let p be the auxiliary gate of a $!$- or $\S$-box of π. Its pax tree contains only contractions and additive auxiliary gates, since exponential auxiliary gates are excluded by the promotion context condition. Similarly, the leaves can be only weakenings and derelictions, axioms being forbidden for the same reason. In particular, if there are any weakening leaves by definition they cannot be infinitary. So every edge of the tree is labeled by some formula $\flat A$, and all structural rules are translated into additive structural rules, therefore not contributing to the size; additive auxiliary gates are also erased, because **LLL** proof-nets do not use additive boxes. Finally, derelictions are too wiped out by the translation.

So a pax tree τ of size t is contracted to a single auxiliary gate, followed by a number of unary why not nodes which is at most equal to the number of leaves of τ, which is of course less than or equal to t. So in the end the size does not grow at all, and part b is proved. $\square$

After executing Girard's "round-by-round" procedure on $\pi^\bullet$, we have at our disposal an **LLL** proof-net σ which, in the lazy case, is cut-free. Translating form **LLL** to **LLL*** is readily done:

Definition 11 (Proof-net translation from $\mathbf{LLL}^*$ to LLL) *If σ is an* **LLL** *proof-net of conclusions*

$$\Gamma; [\Sigma]; \Delta_1; \ldots; \Delta_n$$

where the Δ_i are additive blocks and $[\Sigma]$ is a sequence of discharged formulas, we define its **OLL** *translation $\sigma_\bullet$ as a proof-net of conclusions*

$$\Gamma, ?\Sigma, ?\Delta_1, \ldots, ?\Delta_n$$

If $n = 0$, then $\sigma_\bullet$ will be in $\mathbf{LLL}^*$. Our translation works as follows:

- *Identity links (axioms and cuts) and logical (multiplicative and additive) links are translated to their respective* **OLL** *counterpart. Boxes are used for the translation of &-nodes.*

- *For the additive weakening we have two cases. The first one is the derivation of $\Gamma; A, B$ from $\Gamma; A$, in which an additive block is "created". We simulate this by adding a dereliction to the conclusion A (thus having the new conclusion $?A$) and a weakening of conclusion $?B$.*

  The second case is the derivation of $\Gamma; \Delta, A$ from $\Gamma; \Delta$, in which an additive block already exists; by induction we have an **OLL** *proof-net of conclusions $\Gamma, ?\Delta$, to which we simply add a weakening node of conclusion $?A$.*

- *Why not nodes are converted according to their arity: 0-ary nodes become weakenings, unary nodes are simply suppressed and n-ary nodes with $n \geq 2$ are converted into clusters of contractions.*

- *$!$-boxes and $\S$-boxes are translated as follows. We consider the more complicated case of a $\S$-box, the case of a $!$-box being practically identical. If the content of the box is a proof-net of conclusions $\Gamma; \Delta_1; \ldots; \Delta_m; A_1; \ldots; A_n$, where the Δ_i are additive blocks, we know by induction that we have a corresponding* **OLL** *proof-net of conclusions $\Gamma, ?\Delta_1, \ldots, ?\Delta_m, A_1, \ldots, A_n$. Then, we apply a dereliction node to each of the conclusions in Γ, and build immediately after a box with conclusions $?\Gamma, ?\Delta_1, \ldots, ?\Delta_m, \S A_1, \ldots, \S A_n$, which correctly yields an $\mathbf{LLL}^*$ proof-net.*

Checking that the definition we just gave yields correct $\mathbf{LLL}^*$ proof-nets (i.e. proof-nets that respect inhibition and both context conditions) is easy and left to the reader. The counterpart to lemma 10 also holds trivially from the definition:

Lemma 11 *If σ is an* **LLL** *proof-net of size s, then its translation $\sigma_\bullet$ can be generated in time $O(s)$.*

So we can state the **FP**-soundness of $\mathbf{LLL}^*$:

Proposition 12 *If π is a* lazy $\mathbf{LLL}^*$ *proof-net of size s and exponential depth d, then its normal form π' can be calculated in time $O(s^{2^{\max\{d,1\}}})$.*

Proof. We start with π of size s and depth d. We apply the first renaming, and get $\pi^{\mho_0}$, which takes time $O(s)$ and does not touch either size or depth. Then we apply the second renaming to get $\pi^{\mho}$, still leaving size and depth untouched and taking time $O(s^2)$. Now we can compute in linear time $\pi^\bullet$, which by lemma

10 only multiplies the size by a constant, say k, and obviously does not change the exponential depth. So reducing this **LLL** proof-net with Girard's strategy (which is Church-Rosser and leads to a cut-free proof since we are lazy by hypothesis) takes no more than $3(ks)^{2^d}$ steps (see [Gir98]), and yields an **LLL** proof-net σ of size at most $(ks)^{2^d}$. So, by lemma 11 we take another $O((ks)^{2^d})$ computational steps to translate this into $\sigma_\bullet = \pi'$.

If we want to be pedantic, the total step number would be something like

$$2s + s^2 + 4(ks)^{2^d} = O(s^{2^{\max\{d,1\}}})$$

which is just what we wanted. $\square$

References

[Bai02] Patrick Baillot. **Stratified coherent spaces: a denotational semantics for Light Linear Logic**. To appear in *Theoretical Computer Science*, Special issue on *Implicit Computational Complexity*, 2002.

[DJ99] Vincent Danos and Jean-Baptiste Joinet. **Linear Logic & Elementary Time**. In *Proceedings of ICC '99*, 1999.

[Gir98] Jean-Yves Girard. **Light Linear Logic**. *Information and Computation*, 14(3):175–204, 1998.

[Maz02] Damiano Mazza. *Logica Lineare e Complessità Computazionale*. Tesi di laurea, Università degli Studi Roma Tre, Rome, 2002.

[Rov99] Luca Roversi. **A P-Time Completeness Proof for Light Logics**. In *CSL: 13th Workshop on Computer Science Logic*. LNCS, Springer-Verlag, 1999.

[Ter01] Kazushige Terui. **Light Affine Calculus and Polytime Strong Normalization**. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS-01)*, pages 209–220, Los Alamitos, CA, June 16–19 2001. IEEE Computer Society.