Dissymetrical Linear Logic

Jean-Baptiste Joinet

Dissymetrical Linear Logic

Jean-Baptiste Joinet
IRPhiL, Université Jean Moulin Lyon, Lyon, France *
IHPST, UMR 8590 CNRS-Université Panthéon-Sorbonne, Paris, France †
jean-baptiste.joinet@univ-lyon3.fr

This paper is devoted to design computational systems of linear logic (i.e. systems in which, notably, the non linear and structural phenomena which arise during the cut-elimination process are taken in charge by specific modalities, the exponentials: ! and ?). The systems designed are "intermediate" between Intuitionistic LL and Classical LL. Methodologically, the focus is put on how to break the symmetrical interdependency between ! and ? which prevails in Classical LL – and this without to loose the computational properties (closure by cut-elimination, atomizability of axioms). Three main systems are designed (Dissymetrical LL, semi-functorial Dissymetrical LL, semi-specialized Dissymetrical LL), where, in each of them, ! and ? play well differentiated roles.

1 Introduction

1.1 Preliminaries: Linear Logic's standard definitions, notations and terminology

In all of the present paper, the formulas we consider could have been the standard ones of full Second Order Classical Linear Logic [3]. However, for sake of brevity, and because we will mainly focus on the *exponentials* of Linear Logic (modalities ! and ?), we limit us to recalling the propositional fragment with $\multimap$ (linear implication), $\neg$ (negation), and the two exponentials ! and ?. The formulas that we actually consider are thus the ones generated by $A :: X \mid \neg A \mid A \multimap A \mid !A \mid ?A$. And, among the rules of the full bilateral sequent calculus for Second Order Classical Linear Logic[1] (CLL), we only explicit the following ones (where Γ, Δ etc denotes multi-sets of formulas):

Identity rules

$$\frac{}{A \vdash A}\,\mathsf{ax} \qquad\qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\,\mathsf{cut}$$

Introduction rules for propositional connectives

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \multimap B, \Delta}\,\multimap \qquad \multimap\,\frac{\Gamma \vdash A, \Delta \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \multimap B \vdash \Delta, \Delta'} \qquad\qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\,\neg \qquad \neg\,\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}$$

Introduction rules for exponentials

$$\frac{!\Gamma \vdash A, ?\Delta}{!\Gamma \vdash !A, ?\Delta}\,\mathsf{!\text{-}box} \qquad \mathsf{!\text{-}der}\,\frac{\Gamma, A \vdash \Delta}{\Gamma, !A \vdash \Delta} \qquad\qquad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash ?A, \Delta}\,\mathsf{?\text{-}der} \qquad \mathsf{?\text{-}box}\,\frac{!\Gamma, A \vdash ?\Delta}{!\Gamma, ?A \vdash ?\Delta}$$

Structural rules

$$\mathsf{!\text{-}ctr}\,\frac{\Gamma, !A, !A \vdash \Delta}{\Gamma, !A \vdash \Delta} \qquad \mathsf{!\text{-}w}\,\frac{\Gamma \vdash \Delta}{\Gamma, !A \vdash \Delta} \qquad\qquad \frac{\Gamma \vdash ?A, ?A, \Delta}{\Gamma \vdash ?A, \Delta}\,\mathsf{?\text{-}ctr} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash ?A, \Delta}\,\mathsf{?\text{-}w}$$

*Supported by the Bourgeon Research Program "Abstraction & Classification: Computation & Types", Univ. Lyon 3.
†Supported by the Research Program "Geometry of Algorithms", Agence Nationale de la Recherche.
[1] In the literature, Classical Linear Logic is commonly simply called Linear Logic and noted LL.

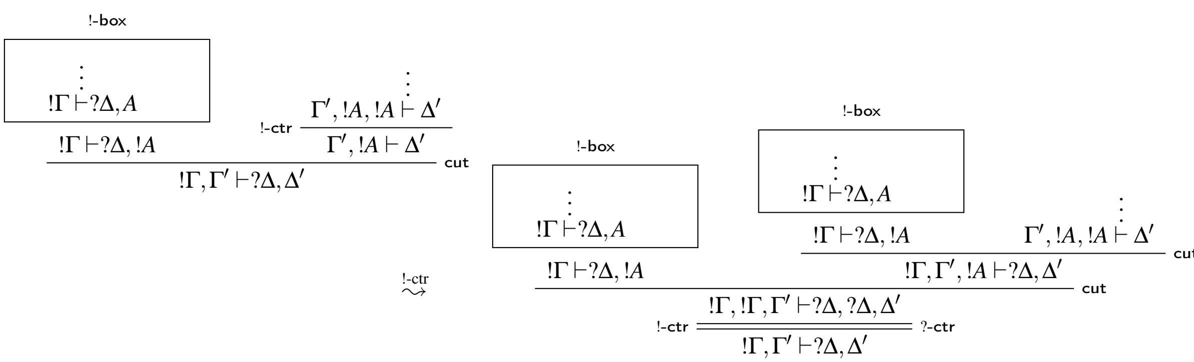
Submitted to:
TLLA-LINEARITY-2022

In all those rules, the multi-sets depicted using capital greek letters ($\Gamma, !\Gamma, \Gamma', \Delta, ?\Delta$ etc) are called *contexts*. Among formula occurrences figured in a rule, we distinguish (using a fairly standard terminology): (a) the *passive* ones, namely the ones belonging to a context (in the particular case of box-rules, the passive occurrences present in the conclusion sequent are also called *auxiliary doors* of the box); (b) the *main* ones, namely the one(s) present in the conclusion sequent of the rule, which are not element of a context (in the particular case of box-rules, the main occurrence is also called *the main door* of the box); (c) the *active* ones, namely all of the remaining formula occurrences (in the particular case of a cut, the two active occurrences are also called *the cut formulas*). An *exponential rule* is either a rule introducing an exponential or a structural rule.

From now on, for pedagogical reasons which will be clear soon, we will often represent the two context dependent rules (right ! and left ? introduction) using a "boxed" notation as below (and not using simply a line as above) – whence the names !-box and ?-box respectively given to them[2]:

The advantages of such a "boxed" representation become patent when one considers the (standard) cut-elimination process (reduction) and, specifically, what happens to boxes when one performs the elementary reduction steps (e.r.s.) properly involving them. Those e.r.s. are the ones to be performed when: 1/ one of the cut-formulas is "the main door" of a box (below, for sake of brevity, we only present the cases where that box is a !-box – the missing *ers*, with ?-boxes, may be easily recovered, since their design is the same up to the right/left symmetry) and 2/ the other one is either (a) main in a contraction, a weakening or a dereliction (the six corresponding e.r.s. are noted $\overset{!\text{-ctr}}{\rightsquigarrow}, \overset{!\text{-w}}{\rightsquigarrow}, \overset{!\text{-de}}{\rightsquigarrow}, \overset{?\text{-ctr}}{\rightsquigarrow}, \overset{?\text{-w}}{\rightsquigarrow}, \overset{?\text{-der}}{\rightsquigarrow}$) or (b) auxiliary door of a box-rule (e.r.s. notation[3]: $\underset{[!\text{-box}]}{\overset{!\text{-aux}}{\rightsquigarrow}}$, $\underset{[?\text{-box}]}{\overset{!\text{-aux}}{\rightsquigarrow}}$, $\underset{[!\text{-box}]}{\overset{?\text{-aux}}{\rightsquigarrow}}$, $\underset{[?\text{-box}]}{\overset{?\text{-aux}}{\rightsquigarrow}}$)[4].

1. Definition of the $\overset{!\text{-ctr}}{\rightsquigarrow}$ elementary reduction step:

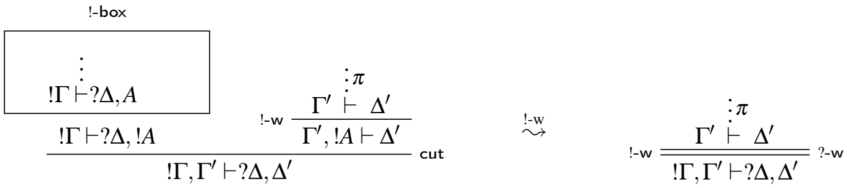


[2]In Linear Logic, the !-box and ?-box rules are usually called the "promotions rules". The rules !-der and ?-der are the "dereliction rules", !-ctr and ?-ctr the contraction rules, !-w and ?-w the weakening rules. The use of "boxes" comes from the proof-nets representation for Linear Logic proofs – a quotient over sequent calculus CLL-derivations [4]. Actually, from now on, the reader has to read our sequent calculus proofs as being simply a convenient *notation* for the (unique) corresponding proofnet.

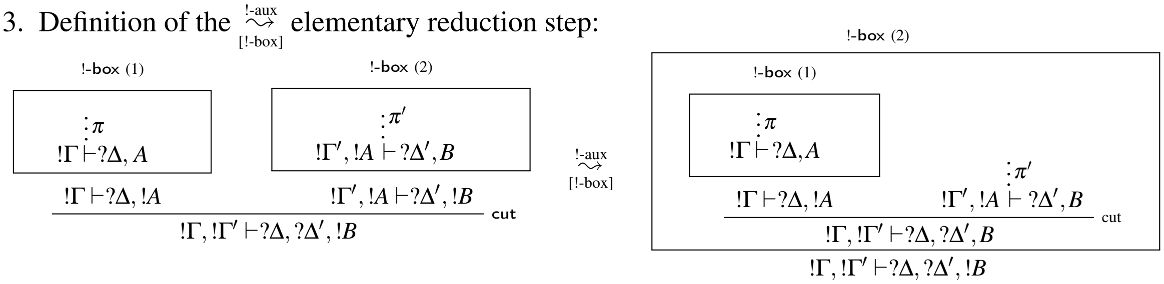
[3]The mention over $\rightsquigarrow$ indicates which exponential prefixes the cut formula which is an auxiliary door (and which by the way also prefixes the other cut formula); the mention under $\rightsquigarrow$ indicates of which kind of box (!-box/?-box) that cut formula is an auxiliary door.

[4]We thus leave unpresented the trivial cases where the second cut-formula is passive in a rule other than a box-rule (cases which disappear in Proof-nets and can thus be neglected) or is active in an identity axiom (case which is not specific to exponentials).

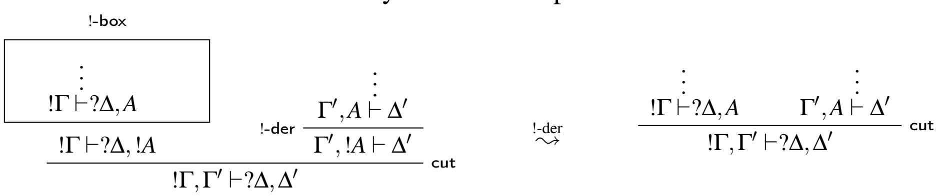
2. Definition of the $\overset{!\text{-w}}{\rightsquigarrow}$ elementary reduction step:



3. Definition of the $\underset{[!\text{-box}]}{\overset{!\text{-aux}}{\rightsquigarrow}}$ elementary reduction step:



4. Definition of the $\overset{!\text{-der}}{\rightsquigarrow}$ elementary reduction step:



A first advantage of the "boxed" representation is that it renders visible the *non local* nature of processes involving exponentials: boxes (*and their content*) are duplicated by contractions through $\overset{!\text{-ctr}}{\rightsquigarrow}$, they are erased by weakenings through $\overset{!\text{-w}}{\rightsquigarrow}$, they are swallowed by other boxes through $\overset{!\text{-aux}}{\rightsquigarrow}$ . . . and this up to the moment where, maybe, through the $\overset{!\text{-der}}{\rightsquigarrow}$, the boxing instruction is abandoned (the box itself disappears, but *not its content*).
Since boxes naturally induce an ordering (boxes into boxes into boxes etc) and a notion of depth (the minimal number of boxes to be crossed to reach the external world), the "boxed" representation renders also visible that, during the cut-elimination process, the depth of formulas/rules may increase (through $\overset{!\text{-aux}}{\rightsquigarrow}$) as well as decrease (through $\overset{!\text{-der}}{\rightsquigarrow}$).

1.2 About !/? interdependency

In CLL, because of the constraint over the context in box-rules, neither of the two exponentials has an autonomous existence. For instance, introducing a ! on the right side of a sequent with multiple conclusions *requires* that the other formulas in the right part of the sequent are prefixed with a ?-exponential. Hence, to be used, such an introduction rule for ! requires the existence of introduction rules for ?. Moreover, the calculus being fully symmetrical, the converse dependency also prevails. The two exponentials are thus *interdependent*.

A crucial point, then, is that such a "static" interdependency of exponentials also entails their *dynamic interdependency*: typically, when a !-contraction duplicates a box (hence a !-box) through a $\overset{!\text{-ctr}}{\rightsquigarrow}$ step, new ?-contractions are created (over the ?-prefixed auxiliary doors of the duplicated !-box); so, later on in the cut-elimination process, those ?-contractions may well come to duplicate ?-boxes (this time, through a $\overset{?\text{-ctr}}{\rightsquigarrow}$ step). So, in the chain of events, !-contractions generally induce non linear effects (duplication,

erasure) over ?-boxes. The interdependency leads in a way to mix up the roles of ? and !.
Another symptom of that "confusion of roles" (which is also an effect of that interdependency) is that no auxiliary door is "by essence" auxiliary door of, say, a ?-box (or of a !-box as well). Indeed, an auxiliary door of a ?-box (resp. a !-box) may well "become" auxiliary door of a !-box (resp. ?-box), because of the e.r.s. $\underset{[?\text{-box}]}{\overset{!\text{-aux}}{\rightsquigarrow}}$ (resp. $\underset{[!\text{-box}]}{\overset{?\text{-aux}}{\rightsquigarrow}}$).

Actually, those features appear problematic only if one wish ? and ! play well differentiated roles. After all, a corollary of the full right/left *symmetries of the rules* is that CLL enjoys everywhere *de Morgan* duality[5] and this is also the case for the two exponentials ($!/?$): $\neg !A$ (resp. $\neg ?A$) is provably equivalent to $?\neg A$ (resp. $!\neg A$) in CLL. So, seen through the eyes of duality, the underlined "confusion" may be simply understood as "redundancy". And, indeed, it is common when it comes to present Classical Linear Logic through Sequent Calculus, to soon switch to a presentation where the set of formulas is quotiented by *de Morgan* equivalences, where the sequents are single sided ($\vdash \Gamma$), where the identity constraints of "Identity rules" are replaced by duality constraints and where, concerning exponentials, there is no more ways to "confuse roles", simply because roles are now univoquely attributed: all the boxes are !-boxes (having only ?-auxiliary doors) and all the contractions (weakenings, derelictions) are ?-contractions (?-weakenings, ?-derelictions). Such a way for obtaining "de-confusion" is however possible only when a perfect, full symmetry prevails: in Classical Linear Logic.

Nevertheless, another way to avoid that "confusion of roles" would be *a priori* possible: by *breaking the symmetry* (hence loosing the duality $!/?$), we could make the symmetrical interdependency of ? and ! cease, so that each of the resulting (no more dual) exponentials ? and ! would eventually becomes independent (at least with some respects) and play a proper, well differentiated role. The question then is whether *computational* such non symmetric subsystems of CLL exist, i.e. can be designed.

A preliminary observation here is that, although being a dissymmetrical (computational[6] and by the way powerful[7]) fragment of CLL, Intuitionistic Linear Logic (ILL) is not a fragment of the kind aimed at. Defined as the fragment of CLL where sequents have at most one conclusion, it clearly does break the symmetry. But it does it in a so drastic way, that half of the baby is gone with the bathwater: there is no more interdependency, but just because only one exponential survives to the treatment. Indeed, to keep considering the (right) ?-ctr in a single conclusion sequents frame would be of no use (and in presence of neutrals – whose presentation has been omitted, for sake of brevity – a similar remark can be done for ?-w). And with no right structural rules, the need for ?-exponentials completely ceases from the dynamic viewpoint. *De facto*, the language of ILL only includes the ! exponential. In that frame, ! becomes indeed a properly autonomous connective and the symmetrical interdependency $?/!$ indeed vanished, but this just because ? itself disappeared.

The rest of the paper is devoted to examine how to break the symmetry of CLL in a less drastic way than ILL does, in order to design computational fragments of CLL where the $?/!$ *symmetric interdependency* we observed will not prevail anymore (even if some non symmetric dependencies between them will subsist), so that ? and ! will be able to play specific, well differentiated roles. In a sense, our aim is thus to find intermediate computational Linear Logics "between" ILL and CLL.

---

[5]Each connective/quantifier of CLL comes with its dual : $\otimes/\parr$, $\&/\oplus$, $1/\bot$, $0/\top$, $\forall/\exists$ are all pairs of duals (some of them being here cited even if their rules have not been presented)

[6] ILL is closed by expansion of identity axioms (i.e. identity axioms $\overline{A \vdash A}$ may be canonically proved from "atomic" identity axioms $\overline{X \vdash X}$, by using only the rules introducing the connectives involved in A.

[7] Functions whose totality is provable in second order arithmetic are all representable in second order ILL (Girard's System F can faithfully be embedded in it).

2 Dissymetrical Linear Logic (DLL)

Definition 1 *Full 'Dissymetrical Linear Logic',* DLL*, is the system one got by replacing (in* CLL*)* $\frac{!\Gamma \vdash A, ?\Delta}{!\Gamma \vdash !A, ?\Delta}$!-box *by* $\frac{!\Gamma \vdash A}{!\Gamma \vdash !A}$!-box *(other* CLL *rules being unchanged).*

Proposition 2 *The system* DLL (i) *is closed by cut-elimination (the potentially problematic e.r.s.* $\overset{?\text{-}aux}{\underset{[!\text{-}box]}{\rightsquigarrow}}$ *never applies);* (ii) *is closed by expansion of identity axioms;* (iii) *is thus a computational fragment of* CLL *(hence is strongly normalizing – and confluent for the additive free fragment).* (iv) *It proves the same* de Morgan *equivalences than* CLL*, but for the* $!/?$ *duality (*$\neg !X \vdash ?\neg X$ *is not cut free provable in* DLL*);* (v) *thus has a strictly weaker expressive power than* CLL*'s one (at the provability level);* (vi) *is evidently at least as powerful as System F (since* ILL *is a sub-system of* DLL*).*

Remark 3 *As for* ?*:* (i) *at the static level,* ? *is not autonomous, it depends on* ! *(introducing* ? *on the left hand side of a sequent generally requires that* ! *have been introduced);* (ii) *that static dependence results in an absence of dynamic autonomy: a* ?-ctr *(resp. a* ?-w*) may "cause" the duplication (resp. erasing) of a* ?*-box, but also of a* !*-box;* (iii) *auxiliary doors of* ?*-boxes (be them prefixed by* ? *or by* !*) are proper to* ?*-boxes: during the cut-elimination process, an auxiliary door of a* ?*-box, never "becomes" auxiliary door of a* !*-box (contrary to what happens in* CLL *as we saw in subsection 1.2). In the next sections, we will take advantage of this property, to treat the* ?*-contexts in specific ways.*

Remark 4 *As for* !*:* (i) *at the static level,* ! *is however autonomous (it does not require the presence of* ?*-exponentials to be introduced);* (ii) *that static autonomy results in a dynamic autonomy: a* !-ctr *(resp. a* !-w*) can only duplicate (resp. erase)* !*-boxes;* (iii) !*-auxiliary doors are not "proper" to* !*-boxes: during the cut-elimination process, an* !*-auxiliary door of a* !*-box may well "become" auxiliary door of a* ?*-box.*

Observe that remark 4 underlines that the dependence $?/!$ observed in remark 3 is no more a symmetrical interdependence (as the one prevailing in CLL). In the next two sections, we use remark 3-(iii) to reinforce the germinal dissymmetry introduced by DLL. We will examine two main ways to perform such a reinforcement: the first one by considering what we call a semi-functorial ?-promotion/box rule (next section); the second by considering ?-box rules using specialized exponentials [2].

3 Semi-functorial Dissymetrical Linear Logic (system sfDLL)

To start with, let us recall the (fully) functorial versions of promotions/boxes[8]:

$$\frac{\Gamma \vdash A, \Delta}{!\Gamma \vdash !A, ?\Delta}\ !\text{-fbox} \qquad\qquad \frac{\Gamma, A \vdash \Delta}{!\Gamma, ?A \vdash ?\Delta}\ ?\text{-fbox}$$

The system (fCLL, for *functorial Classical Linear Logic*[9]) obtained by replacing in CLL the rules !-box and ?-box by their functorial version !-fbox and ?-fbox is (or may be seen as) a computational (and duality enjoying) fragment of CLL. By the way, this is still the case, when one moreover gets rid of !-der and ?-der[10].

[8] They have been notably considered in [5] and [1].

[9] The word *classical* connotes the symmetrical forms of sequents and rules; it should not be understood as suggesting that the system is computationnally as powerful as classical logic, which is false.

[10] In [1] (and already in the unpublished work of the same authors which inspired [5] – cf. p.176), those systems were respectively called TLL and KLL in reference to the famous principles T and K of modal logic. The computational stakes of those choices concerning boxes and derelictions rules are enlightened by our remark (page 3) about "depth": with no derelictions, "depth" never decreases; with f-boxes in place of standard boxes, "depth' never increases. When the two options are combined, the proofs and the computations are so to speak *stratified* (the computation is internal to each layer; layers never interact). This phenomena is the main source of the taming of computational complexity in ELL and LLL [5].

The "symmetrical interdependency" between ! and ? that we observed for CLL persists in those two systems (fCLL and fCLL with no derelictions) for the resulting dual exponentials. In our quest for more autonomous exponentials, we could start by considering the (fully) *functorial Dissymetrical Linear Logic* fDLL, i.e. the system one gets by replacing (in DLL) the two box rules by their functorial version $\frac{\Gamma \vdash A}{!\Gamma \vdash !A}$!-f-box and $\frac{\Gamma, A \vdash \Delta}{!\Gamma, ?A \vdash ?\Delta}$?-f-box (all other DLL rules being unchanged). Even if it is computational, we will leave fDLL aside (as well as its version with no derelictions), because of the weakness of its comptutational expressive power[11].

Let us observe that it would not be possible, starting from CLL (neither from DLL, actually), to choose to have one of the two box-rules (e.g. the ?-box) being functorial while the other one (the !-box) would remain standard. Indeed, as we saw in remark 3-(iii), the static of CLL leads to a dynamic "confusion of the roles" of ? and ! (a complete one in CLL, a partial one in DLL), which in both cases prevents those systems to be closed through $\underset{[?\text{-fbox}]}{\overset{!\text{-aux}}{\rightsquigarrow}}$.

In the next section, we examine possible ways to nevertheless introduce some functoriality in (variants of) DLL.

Definition 5 *We call semi-functorial Dissymetrical Linear Logic (*sfDLL*), the system one gets by replacing in* DLL *the* ?-box *rule, by its semi-functorial version:* ?-sf-box $\frac{!\Gamma, A \vdash \Delta}{!\Gamma, ?A \vdash ?\Delta}$ *(all other rules of* DLL *being kept).*

Proposition 6 sfDLL *is a computational sub-system of* DLL*, strongly normalizing (and confluent for the additive free fragment). By the way, it is also the case for* sfDLL *without the* ?*-dereliction rule.*

From the provability point of view, the expressivity of sfDLL is lower than the one of DLL (e.g. $??X \vdash ?X$ is not provable in sfDLL). If one moreover drops the ?-der rule, one also looses $X \vdash ?X$.

Expansion of identity axioms in sfDLL is easy to check. As for closure by cut elimination, only the potentially problematic steps have to be checked, namely the one where a box "swallows" a box (for all other kinds of e.r.s. do not differ depending on the kind of boxes involved, if any). Strong Normalization and Church-Rosser properties are corollaries of the fact that CLL enjoys them (knowing that sfDLL is a fragment of it).
As for sfDLL without the ?-dereliction rule, it suffice to observe that it is neither used to prove *expansion of identity axioms*, nor to prove the *closure by cut-elimination*.

In the next section, we show that once the symmetry !/? is broken as in DLL, one may also assign differentiated, specialized roles to ? and !.

4 Semi-specialized Dissymetrical Linear Logic (ssDLL)

In [2], the possibility to consider "specialized"-exponentials $?_w$ and $?_c$, i.e. "weak" exponentials dedicated to specific non linear effects ($?_w$ being specialized in erasures, $?_c$ in duplications) was considered. Naturally, introducing such exponentials imposes that the "boxes" subject to the corresponding non linear effects are ready for that (i.e., using the Proof-net terminology, that the auxiliary doors of those boxes are themselves endowed with the required ability). We continue to use the notation "!" for !-exponentials endowed with the "full ability" (i.e. both weakening and contraction, as usual).

[11] Logics like 2^d order LJ (hence System F) or LK are not uniformly translatable in functorial systems. So, even if trying to understand their expressive power as a type system (i.e. at the computational complexity level) may be an interesting target, one could prefer to look for a less restricted system. Typically, we would like to have functorial features, in a system *extending* ILL (i.e. able to interpret System F) as we indeed do.

In the system below, the relative independence of both exponentials allows to attribute them well differenciated roles: the ! plays the same usual role than in DLL, where ? is restricted to specialized roles.

The rules of the "semi-specialized Dissymmetrical Linear Logic" (ssDLL) are the following:

$$
\text{!-der}\frac{\Gamma, A \vdash \Delta}{\Gamma, !A \vdash \Delta}
\quad
\text{!-box}\frac{!\Gamma \vdash A}{!\Gamma \vdash !A}
\qquad
\underset{w}{?}\text{-box}\frac{!\Gamma, A \vdash \underset{w}{?}\Delta}{!\Gamma, \underset{w}{?}A \vdash \underset{w}{?}\Delta}
\quad
\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \underset{w}{?}A, \Delta}\underset{w}{?}\text{-der}
\qquad
\underset{c}{?}\text{-box}\frac{!\Gamma, A \vdash \underset{c}{?}\Delta}{!\Gamma, \underset{c}{?}A \vdash \underset{c}{?}\Delta}
\quad
\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \underset{c}{?}A, \Delta}\underset{c}{?}\text{-der}
$$

$$
\text{!-ctr}\ \frac{\Gamma, !A, !A \vdash \Delta}{\Gamma, !A \vdash \Delta}
\quad
\text{!-w}\ \frac{\Gamma \vdash \Delta}{\Gamma, !A \vdash \Delta}
\qquad
\frac{\Gamma \vdash \Delta}{\Gamma \vdash \underset{w}{?}A, \Delta}\ \underset{w}{?}\text{-w}
\qquad
\frac{\Gamma \vdash \underset{c}{?}A, \underset{c}{?}A, \Delta}{\Gamma \vdash \underset{c}{?}A, \Delta}\ \underset{w}{?}\text{-ctr}
$$

Proposition 7 ssDLL *is (or rather : can be seen as) a computational sub-system of* DLL*, strongly normalizing (and confluent, for the additive free fragment).*

To finish with, let us observe that the semi-functoriality restriction (studied in section 3) and the semi-specialization restriction just presented are fully compatible. For instance, one could well consider (without losing the cut-elimination property) a system where one would replace the exponential rules $\underset{w}{?}$ and $\underset{c}{?}$ above by:

$$
\underset{w}{?}\ \frac{!\Gamma, A \vdash \Delta}{!\Gamma, \underset{w}{?}A \vdash \underset{w}{?}\Delta}
\qquad\qquad
\underset{c}{?}\ \frac{!\Gamma, A \vdash \Delta}{!\Gamma, \underset{c}{?}A \vdash \underset{c}{?}\Delta}
$$

Note that, in that last case (actually, as in the case of fully functorial Classical Linear Logic or fully Functorial Dissymetrical Linear Logic or even sfDLL), adding the corresponding dereliction rules is not compulsory (they are not needed for expansion of identity axioms).

5 Conclusion: future works and applications

The systems designed in the present paper are all computational systems stronger than ILL but weaker than CLL. This suggests two main kinds of applications for them.

As they all extend second order ILL (in which System F can be represented) by introducing the ?-modality in charge of right structural rules (a first step toward "classical logic"), they could be used to capture or classify specific "classical algorithms" (weaker however than the full ones, in the spirit of implicit computational complexity approaches) or at least specific "classical" strategies (e.g. Thomas Ehrard's Call-by-push-value).

As they are intermediate linear logics "between ILL and CLL", they could be used to embed/interpret in Linear Logic, intermediate logics "between intuitionistic logic and classical logic" (or even to embed/interpret multi-conclusions formulations of intuitionistic logic).

References

[1] Vincent Danos & Jean-Baptiste Joinet (2003): *Linear Logic and Elementary time*. *Information and Computation* (183), pp. 123–137, doi:http://dx.doi.org/10.1016/S0890-5401(03)00010-5.

[2] Vincent Danos, Jean-Baptiste Joinet & Harold Schellinx (1993): *The structure of exponentials: uncovering the dynamics of Linear logic derivations*. In Georg Gottlob, Alexander Leitsch & Daniele Mundici, editors: Computational Logic and Proof Theory *(proceedings of the 3rd Kurt Gödel Colloquium, august 1993, Brno, Czech Republic)*, *Lecture Notes in Computer Science* 713, Springer Verlag, Berlin-Heidelberg, Germany, pp. 159–171.

[3] Jean-Yves Girard (1987): *Linear logic*. *Theoretical Computer Science* (50 (1)), pp. 1–102, doi:http://dx.doi.org/10.1016/0304-3975(87)90045-4.

[4] Jean-Yves Girard (1996): *Proof-nets: the parallel syntax for proof theory*. In: *Logic and Algebra*, *Lecture Notes in Pure and Applied Mathematics* 180, Marcel Dekker publisher, New York.

[5] Jean-Yves Girard (1998): *Light Linear Logic*. *Information and Computation* (143), pp. 175–204, doi:http://dx.doi.org/10.1006/inco.1998.2700.