A Formulation of Linear Logic Based on Dependency-Relations

Torben Braüner[1]* and Valeria de Paiva[2]

[1] Centre for Philosophy and Science-Theory
Aalborg University
Langagervej 6
DK-9220 Aalborg East, Denmark
torbenb@hum.auc.dk

[2] School of Computer Science
University of Birmingham
Edgbaston
Birmingham B15 2TT, England
V.DePaiva@cs.bham.ac.uk

Abstract. In this paper we describe a solution to the problem of proving cut-elimination for FILL, a variant of exponential-free and multiplicative Linear Logic originally introduced by Hyland and de Paiva. In the work of Hyland and de Paiva, a term assignment system is used to describe the intuitionistic character of FILL and a proof of cut-elimination is barely sketched. In the present paper, as well as correcting a small mistake in their work and extending the system to deal with exponentials, we introduce a different formal system describing the intuitionistic character of FILL and we provide a full proof of the cut-elimination theorem. The formal system is based on a dependency-relation between formulae occurrences within a given proof and seems of independent interest. The procedure for cut-elimination applies to (multiplicative and exponential) Classical Linear Logic, and we can (with care) restrict our attention to the subsystem FILL. The proof, as usual with cut-elimination proofs, is a little involved and we have not seen it published anywhere.

1 Introduction

Linear Logic was introduced by J.-Y. Girard in [Gir87] and it has attracted much attention from computer scientists, as it is a logical way of coping with resources and resource control. Applications of Linear Logic include linear functional programming, linear logic programming, general theories of concurrency, linguistics, AI and planning. The primary goals of this paper are to provide a system of dependency between formulae for a fragment of Linear Logic and to prove a cut-elimination theorem for this fragment.

* The work presented here was partially carried out while this author was employed by BRICS, Aarhus University. The author is presently supported by the Danish Natural Science Research Council.

Linear Logic comes in various "flavours" and most people know about what is called Classical Linear Logic and about the fragment introduced by Girard and Lafont, called Intuitionistic Linear Logic. Classical Linear Logic has two extra connectives, when compared with Intuitionistic Linear Logic: The multiplicative disjunction $\invamp$ and the modality ?, called by Girard *why not*.

We are here concerned with Full Intuitionistic Linear Logic (FILL), a variant of Linear Logic, introduced by Hyland and de Paiva [HdP93] with the aim of giving a syntactic reconstruction of their previous semantical work on Dialectica categories [dP88]. Dialectica categories came out of Hyland's insight that an internal categorical description of Gödel's Dialectica Interpretation was possible, but instead of providing a model of Intuitionistic Logic, as expected, the dialectica categories provided (a collection of) models of Linear Logic.

The system FILL has all the connectives of Classical Linear Logic. But they behave in a more intuitionistic way: To begin with, the logical connectives are all independent, that is, they are not interderivable, as they are in Classical Linear Logic. (This is analogous to the situation concerning the relationship between Intuitionistic Logic and Classical Logic. In Classical Logic all the connectives can be expressed in terms of implication and negation, whereas in Intuitionistic Logic, conjunction, disjunction and implication are all independent connectives. Only negation is defined in terms of implication and the unit for disjunction, called *falsum*.) In FILL the multiplicative conjunction, the disjunction and linear implication are all independent; only the linear negation $A^{\perp}$ is defined as $A \multimap \perp$ and it is not an involution. Thus we have $A \vdash A^{\perp\perp}$ but not vice-versa, as is the case in Classical Linear Logic. Moreover, we can extend the intuitive Brouwer-Heyting-Kolmogorov interpretation to deal with proofs in FILL.

The syntactical reconstruction in [HdP93] was motivated by Schellinx's observation that the original system did not satisfy cut-elimination [Sch91]. But the presentation in [HdP93] had a mistake and the system as presented, still did not enjoy the cut-elimination property. In this paper - which is a revised version of the technical report [BdP96] - we correct the mistake in the original paper, but more importantly we also introduce a different formal way of describing the intuitionistic character of FILL by providing a notion of dependency between formulae occurrences within a proof. This is accomplished by defining a dependency-relation between formula occurrences in the end-sequent of a given proof. This new notion enables us to give a straightforward proof of cut-elimination.

The work presented here was originally motivated by Girard's work on Light Linear Logic, [Gir94] and we hope to pursue this line of enquiry on future work. For the time being we simply note that we could add (naive) set comprehension (with suitable introduction and elimination rules) to our basic multiplicative system and this would preserve the cut-elimination result. Cut-elimination is not preserved when the exponentials interact with set comprehension; this situation is investigated in [Shi94].

In this paper we first introduce our formal system describing the intuitionistic character of FILL. We also discuss a few examples. Then we need to prove various results making explicit how the process of cut-elimination interacts with the notion of dependencies between formulae. After that a cut-elimination procedure for multiplicative Classical Linear Logic is given, and we can (with care) restrict our attention to the subsystem FILL. Our proof of cut-elimination for Classical Linear Logic is along the lines of the proof of cut-elimination for Classical Logic given in [GLT89] where we have taken the linear context into account. The proof is a little involved and we have not seen it published anywhere. We conclude describing some possible applications within the area of linear functional programming.

Related Work

First of all, our approach should be compared to the system of [HdP93] where a sequent

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

is decorated with terms

$$x_1 : A_1, ..., x_n : A_n \vdash t_1 : B_1, ..., t_m : B_m$$

with aim of imposing an appropriate condition on the implication right rule. This system is comparable to ours by saying that B_j depends on A_i iff the variable x_i occurs free in t_j. The dependencies induced by the cut-rule of [HdP93] then correspond to ours. But some of the other rules introduce unnecessary dependencies which result in the failure of cut-elimination. For example, the par left rule of [HdP93] makes every consequent formula occurrence dependent on the formula introduced. When in the process of cut-elimination a cut is pushed upwards such that it commutes with a par left rule, that is, where a proof

$$\dfrac{\Gamma_1 \vdash A, \Delta_1 \qquad \dfrac{A, \Gamma_2, B \vdash \Delta_2 \qquad \Gamma_3, C \vdash \Delta_3}{A, \Gamma_2, \Gamma_3, B \parr C \vdash \Delta_2, \Delta_3}}{\Gamma_1, \Gamma_2, \Gamma_3, B \parr C \vdash \Delta_1, \Delta_2, \Delta_3}$$

is replaced by the proof

$$\dfrac{\dfrac{\Gamma_1 \vdash A, \Delta_1 \qquad A, \Gamma_2, B \vdash \Delta_2}{\Gamma_1, \Gamma_2, B \vdash \Delta_1, \Delta_2} \qquad \Gamma_3, C \vdash \Delta_3}{\Gamma_1, \Gamma_2, \Gamma_3, B \parr C \vdash \Delta_1, \Delta_2, \Delta_3}$$

this might result in new dependencies invalidating applications of the implication right rule lower down in the involved proof-tree. Such a step is perfectly unproblematic in our system but in the system of [HdP93] it can take a valid proof to one which is invalid - and at the level of terms it can take a typable term to one which is untypable. See the account given in [Bie96]. It is possible to avoid the unnecessary dependencies by imposing appropriate side-conditions on the problematic rules as pointed out by Bellin, cf. [Bie96]. However, the side-conditions are somewhat involved. On the other hand, the dependency-relations of our formulation of FILL captures exactly what is needed, and moreover, this formulation enables cut-elimination to be proved in a straightforward manner.

The mistake in the system by Hyland and de Paiva has also been corrected, independently, by Bierman, [Bie96], and Bellin, [Bel96]. The approach of Bierman [Bie96] using patterns is problematic as some intuitively valid FILL proofs, which indeed are valid in the sense of the present paper, are not provable in his system. For example, the proof

$$\dfrac{\dfrac{}{A \vdash A} \qquad \dfrac{\dfrac{}{A \vdash A}}{A \vdash A, \bot}}{\dfrac{A \vdash A, \bot}{\vdash A \multimap A, \bot}}$$

is valid in our system, but it is not in Bierman's.

Bellin's work [Bel96] uses proofnets; also his system includes the *Mix* rule, which ours does not. Furthermore, exponentials are omitted in his system. The FILL-condition on a proofnet is expressed by exploiting a notion of path in a net. His paper shows that sequent proofs can be represented as two-sided proofnets. He also proves a sequentialisation theorem saying that each such proofnet corresponds to a set of sequent proofs. No proof of cut-elimination is given.

Also Cockett and Seely [CS96] consider the proof theory of FILL (again omitting exponentials). Their goal was to obtain a categorical coherence result stating that a category generated by formulae and (equivalence classes of) proofnets is a free category in an appropriate sense. The FILL-condition on their notion of proofnets is stated using additional syntax; that is, the implication rule introduces a box which is used to keep track of the context. This seems an unncessary complication. Our approach (and Bellin's as well) dispenses with these extra boxes in implication rules, as the FILL-condition is stated at the meta-level. This gives a cleaner notion of reduction in the sense that it avoids complications related to the extra boxes. Furthermore, Cockett and Seely's restriction simply amounts to the right hand side context being empty, which is unsatisfactory from the point of view of cut-elimination.

2 The Dependency System

We define FILL as a subsystem of Classical Linear Logic, as formulae in FILL are the same as the formulae in Classical Linear Logic, and a proof in FILL is a proof in Classical Linear Logic with a certain intuitionistic property.

Definition 1. Formulae of Classical Linear Logic are defined by the grammar

$$S \; ::= \; S \otimes S \mid I \mid S \parr S \mid \bot \mid S \multimap S \mid !S \mid ?S$$

The metavariables A, B, C range over formulae and Γ, Δ range over lists of formulae.

Definition 2. The inference rules for Classical Linear Logic are given in the left columns of Figures 1 and 2.

The metavariables π, τ will range over derivations as well as proofs. We pay no attention to the usual exchange rules. Note that, unlike most presentations of Classical Linear Logic, we use two-sided sequents, as they allow us to draw the finer distinctions that we need.

To define FILL we need a notion of dependency, relating formulae occurrences. An occurrence of a formula will refer to an occurrence of a formula in a given proof or derivation. By an insignificant abuse of notation we let A, B, C range over occurrences of formulae *as well as* formulae. Also we let Γ, Δ range over sets of occurrences of formulae *as well as* lists of formulae. The actual interpretations are to be determined by the contexts. Note that a set of occurrences of formulae might contain more than one occurrence of the same formula. We use $'$ and $''$ to distinguish between two occurrences of the same formula. Basically we must define, given a proof τ of a sequent $\Gamma \vdash \Delta$ in Classical Linear Logic, when a given formula occurrence in the succedent Δ depends on a given formula occurrence in the antecedent Γ. We shall do this by defining a relation $Dep(\tau)$ between Γ, considered as a set of formula occurrences, and Δ, also considered as a set of formula occurrences. This notion of dependency between formulae occurrences in the sequent, when properly defined, allows us to express the intuitionistic property that characterizes FILL.

With the aim of defining the dependency-relation, we need a generalisation of the usual composition operation on relations.

Definition 3. Assume that r_1 is a relation between the set A_1 and the set B_1 and assume that r_2 is a relation between the set A_2 and the set B_2. We then define the relation $r_1 \star r_2$ between $A_1 \cup (A_2 \setminus B_1)$ and $B_2 \cup (B_1 \setminus A_2)$ as

$$
\begin{aligned}
&[(r_1 \cap (A_1 \times (B_1 \cap A_2))) * (r_2 \cap ((A_2 \cap B_1) \times B_2))] \cup \\
&[r_1 \cap (A_1 \times (B_1 \setminus A_2))] \cup \\
&[r_2 \cap ((A_2 \setminus B_1) \times B_2)]
\end{aligned}
$$

where $*$ is the usual composition of relations.

This definition might seem somewhat involved, but note that $r_1 \star r_2$ is defined as the union of the following three sets:

1. The composition of r_1 with codomain restricted to A_2 and r_2 with domain restricted to B_1,
2. r_1 with codomain restricted to $B_1 \setminus A_2$,
3. r_2 with domain restricted to $A_2 \setminus B_1$.

The operation $\star$ is a generalisation of $*$ in the sense that $r_1 \star r_2 = r_1 * r_2$ whenever $B_1 = A_2$. Given sets A and B, we define the relation $A \bullet B$ between A and B as the Cartesian product $A \times B$. Note that if A and B are singletons, say $A = \{a\}$ and $B = \{b\}$, then composing $A \bullet B$ on the left of an arbitrary relation r having b in the domain amounts to replacing each tuple (b, c) in r by (a, c).

Intuitively we can say that "genuine" dependencies start in axioms, constants do not introduce dependencies and dependencies "percolate" through a proof as expected. With these intuitions in mind, the reader is invited to check the rules in the following definition.

Definition 4. Let τ be a proof in Classical Linear Logic whose end-sequent is $\Gamma \vdash \Delta$. The immediate subproofs of τ are denoted by τ_i. By induction on τ, we define the relation $Dep(\tau)$ between Γ, considered as a set of formula occurrences, and Δ, also considered as a set of formula occurrences. The definition is by cases in accordance with the tables in Figures 1 and 2.

It should be obvious how to track dependencies through implicit applications of exchange rules. Note that the $\star$ operation is associative in the cases of the definition of $Dep(\tau)$ where it is involved more than once. However, it is not associative in general. If the derivation τ ends in the sequent $\Gamma \vdash \Delta$, and B and A are formula occurrences in Γ and Δ respectively, then we say that *A depends on B in τ* iff $(B, A) \in Dep(\tau)$. If τ and σ are two derivations ending in the sequent $\Gamma \vdash \Delta$, we say that the end-sequent of τ *has the same dependencies as* the end-sequent of σ iff we for any formula occurrence C in Γ and any formula occurrence A in Δ have that A depends on C in τ iff A depends on C in σ. Similarly, we say that the end-sequent of τ *has fewer dependencies than* the end-sequent of σ iff we for any formula occurrence C in Γ and any formula occurrence A in Δ have that A depends on C in τ entails that A depends on C in σ. We are now in position to state our definition for the system FILL.

Definition 5. A proof in FILL is a proof in Classical Linear Logic where whenever the rule $\multimap_R$ is applied to a proof τ of $\Gamma, B \vdash C, \Delta$ to obtain $\Gamma \vdash B \multimap C, \Delta$ none of the formulae occurrences in Δ depends on the occurrence of B in τ.

The condition on the implication right rule in the definition of dependency deserves careful discussion. The implication right rule in Classical Linear Logic (like the one in CL) allows any (linear) implications whatsoever. The implication right rule in Intuitionistic Linear Logic enforces the existence of a single conclusion on the sequent. The implication right rule for FILL is more liberal than a single formula in the consequent, but more restricted than the classical linear logic rule. It was to express this subtle situation that we needed the concept of dependencies; but dependencies also make sense in the Classical Linear Logic case. Suppose we are given a proof

$$\dfrac{\begin{array}{c}\vdots\ \tau\\ \Gamma, B \vdash C, \Delta\end{array}}{\Gamma \vdash B \multimap C, \Delta}\ \multimap_R$$

and consider any A in the consequent Δ. If A did not depend on B, nothing will change. If A did depend on B, since by applying the implication right rule we just discharged it, it need not be there. Now, if we look at the formula $B \multimap C$ again we have two cases. Either C did depend on B, and then we

Fig. 1. Definition of $Dep(\tau)$ – The Multiplicatives

$\dfrac{}{A' \vdash A''}\,Ax$	$Dep(\tau) = \{(A', A'')\}$
$\dfrac{\Gamma_1 \vdash B', \Delta_1 \quad \Gamma_2, B'' \vdash \Delta_2}{\Gamma_2, \Gamma_1 \vdash \Delta_2, \Delta_1}\,Cut$	$Dep(\tau) = Dep(\tau_1) \star \{B'\} \bullet \{B''\} \star Dep(\tau_2)$
$\dfrac{\Gamma, B, C \vdash \Delta}{\Gamma, B \otimes C \vdash \Delta}\,\otimes_L$	$Dep(\tau) = \{B \otimes C\} \bullet \{B, C\} \star Dep(\tau_1)$
$\dfrac{\Gamma_1 \vdash B, \Delta_1 \quad \Gamma_2 \vdash C, \Delta_2}{\Gamma_1, \Gamma_2 \vdash B \otimes C, \Delta_1, \Delta_2}\,\otimes_R$	$Dep(\tau) = (Dep(\tau_1) \cup Dep(\tau_2)) \star \{B, C\} \bullet \{B \otimes C\}$
$\dfrac{\Gamma \vdash \Delta}{\Gamma, I \vdash \Delta}\,I_L$	$Dep(\tau) = \{I\} \bullet \emptyset \star Dep(\tau_1)$
$\dfrac{}{\vdash I}\,I_R$	$Dep(\tau) = \emptyset$
$\dfrac{\Gamma_1, B \vdash \Delta_1 \quad \Gamma_2, C \vdash \Delta_2}{\Gamma_1, \Gamma_2, B \parr C \vdash \Delta_1, \Delta_2}\,\parr_L$	$Dep(\tau) = \{B \parr C\} \bullet \{B, C\} \star (Dep(\tau_1) \cup Dep(\tau_2))$
$\dfrac{\Gamma \vdash B, C, \Delta}{\Gamma \vdash B \parr C, \Delta}\,\parr_R$	$Dep(\tau) = Dep(\tau_1) \star \{B, C\} \bullet \{B \parr C\}$
$\dfrac{}{\bot \vdash}\,\bot_L$	$Dep(\tau) = \emptyset$
$\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \bot, \Delta}\,\bot_R$	$Dep(\tau) = Dep(\tau_1) \star \emptyset \bullet \{\bot\}$
$\dfrac{\Gamma_1 \vdash B, \Delta_1 \quad \Gamma_2, C \vdash \Delta_2}{\Gamma_1, \Gamma_2, B \multimap C \vdash \Delta_1, \Delta_2}\,\multimap_L$	$Dep(\tau) = Dep(\tau_1) \star \{B, B \multimap C\} \bullet \{C\} \star Dep(\tau_2)$
$\dfrac{\Gamma, B \vdash C, \Delta}{\Gamma \vdash B \multimap C, \Delta}\,\multimap_R$	$Dep(\tau) = \emptyset \bullet \{B\} \star Dep(\tau_1) \star \{C\} \bullet \{B \multimap C\}$

simply get rid of it; or it did not and in this case the dependencies of $B \multimap C$ are the same as the dependencies of C. One may wonder about the *relevant* aspects of this implication. After all Classical Linear Logic is supposed to be more than a relevant logic, i.e that we should only have linear implications where the antecedent is used to produce the consequent and moreover it must be used exactly once. How can we have (linear) cases where the antecedent abstracted over did not show up in the body. Well, we do have rules within Classical Linear

Fig. 2. Definition of $Dep(\tau)$ – The Modalities

$$\frac{\Gamma, B \vdash \Delta}{\Gamma, !B \vdash \Delta}\,!_L \qquad Dep(\tau) = \{!B\} \bullet \{B\} \star Dep(\tau_1)$$

$$\frac{!\Gamma \vdash B, ?\Delta}{!\Gamma \vdash !B, ?\Delta}\,!_R \qquad Dep(\tau) = Dep(\tau_1) \star \{B\} \bullet \{!B\}$$

$$\frac{!\Gamma, B \vdash ?\Delta}{!\Gamma, ?B \vdash ?\Delta}\,?_L \qquad Dep(\tau) = \{?B\} \bullet \{B\} \star Dep(\tau_1)$$

$$\frac{\Gamma \vdash B, \Delta}{\Gamma \vdash ?B, \Delta}\,?_R \qquad Dep(\tau) = Dep(\tau_1) \star \{B\} \bullet \{?B\}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, !B \vdash \Delta}\,W_L \qquad Dep(\tau) = \{!B\} \bullet \emptyset \star Dep(\tau_1)$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash ?B, \Delta}\,W_R \qquad Dep(\tau) = Dep(\tau_1) \star \emptyset \bullet \{?B\}$$

$$\frac{\Gamma, !B', !B'' \vdash \Delta}{\Gamma, !B \vdash \Delta}\,C_L \quad Dep(\tau) = \{!B\} \bullet \{!B', !B''\} \star Dep(\tau_1)$$

$$\frac{\Gamma \vdash ?B', ?B'', \Delta}{\Gamma \vdash ?B, \Delta}\,C_R \quad Dep(\tau) = Dep(\tau_1) \star \{?B', ?B''\} \bullet \{?B\}$$

Logic that introduce "fake" dependencies, that is, antecedent formulae that no consequent formulae depends on. They are I_L,$\bot_L$ and W_L, for example in the derivation

$$\cfrac{\cfrac{\overline{A \vdash A}}{A, I \vdash A}}{A \vdash I \multimap A}$$

we are allowed to abstract I and A does not depend on it. If we disregard the above mentioned rules that introduce false dependencies, then any antecedent formula would have at least one consequent formula depending on it. This would entail that whenever we have an application of the rule for implication right valid in FILL, then the antecedent abstracted over always shows up in the body. We could have an eager version of the dependency definition for the implication right rule where C would be assumed to depend on B to have the implication $B \multimap C$ valid. But, with the other rules the way they are, we would not obtain cut-elimination with the eager version of the FILL condition. For example, the end-sequent of the proof

$$\dfrac{\dfrac{\dfrac{}{A \vdash A} \qquad \dfrac{}{I \vdash I}}{A, I \vdash A \otimes I} \qquad \dfrac{\dfrac{\dfrac{}{A \vdash A}}{A, I \vdash A}}{A \otimes I \vdash A}}{\dfrac{A, I \vdash A}{A \vdash I \multimap A}}$$

would not be provable by a cut-free proof.

Remark. To explain the different flavour of the FILL implication right rule when compared to the Classical Linear Logic one, we could extend (a little) the Brouwer-Heyting-Kolmogorov interpretation of Intuitionistic Logic to say that a proof of $\Gamma \vdash \Delta$ is a function that maps a proof of each formula in Γ to a proof of one of the formulae in Δ together with a specification of which of the formulae is actually proved. The intuitionistic interpretation of a proof of a formula $B \multimap C$ is a function that maps a proof of B into a proof of C. So given a proof τ of the sequent $\Gamma, B \vdash C, \Delta$ and a proof of each formula in Γ, we get a function that maps a proof of B into a proof of either C or a formula in Δ. By imposing the condition that none of the formulae occurring in Δ are allowed to depend on B in τ, we can assume that our function has the property that either we get a proof of C for any proof of B, or we get a proof of a formula in Δ for any proof of B. This enables us to define a function that maps a proof of each formula in Γ into either a proof of $B \multimap C$ or a proof of a formula in Δ.

Example 1. The system FILL is more intuitionistic than Classical Linear Logic, as for instance the "multiplicative excluded middle" $\vdash A^{\perp} \parr A$ cannot be proved in FILL[3]. The usual Classical Linear Logic proof

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}}{A \vdash \perp, A}}{\vdash A \multimap \perp, A}}{\vdash (A \multimap \perp) \parr A}$$

is not a FILL proof as the only formula in Δ, namely A, depends on the formula A we thought of abstracting over. It is easy to see that this entails that any Classical Linear Logic proof of $A^{\perp\perp} \vdash A$ cannot be a FILL proof. Hence, the sequent $A^{\perp\perp} \vdash A$ is not provable in FILL either. But it is surprisingly the case that if a sequent $\vdash A^{\perp\perp}$ is provable in FILL, then also $\vdash A$ is provable in FILL.

[3] In this and the following non-FILL examples we use cut-elimination, as pointed out by Mathias Kegelmann.

This is so because a cut-free proof of $\vdash A^{\perp\perp}$ has to look as follows

$$\dfrac{\vdash A \qquad \dfrac{}{\perp \vdash \perp}}{\dfrac{(A \multimap \perp) \vdash \perp}{\vdash (A \multimap \perp) \multimap \perp}}$$

So we cannot prove the sequent $\vdash (A^{\perp} \parr A)^{\perp\perp}$ in FILL because we cannot prove $\vdash A^{\perp} \parr A$. This is different from the situation in Intuitionistic Logic where we cannot prove $\vdash \neg A \vee A$ but where we *can* prove $\vdash \neg\neg(\neg A \vee A)$. It is perhaps worth mentioning that the Intuitionistic Logic proof of $\vdash \neg\neg(\neg A \vee A)$ uses contraction, not generally available in FILL, and, it uses the $\vee_R$ rule which is different from $\parr_R$. Another example of a sequent provable in Classical Linear Logic but not in FILL is $A \otimes B \multimap (C \parr D) \vdash (A \multimap C) \parr (B \multimap D)$; the Classical Linear Logic proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A} \qquad \dfrac{}{B \vdash B}}{A, B \vdash A \otimes B} \qquad \dfrac{\dfrac{}{C \vdash C} \qquad \dfrac{}{D \vdash D}}{C \parr D \vdash C, D}}{(A \otimes B) \multimap (C \parr D), A, B \vdash C, D}}{(A \otimes B) \multimap (C \parr D), A \vdash C, B \multimap D}}{(A \otimes B) \multimap (C \parr D) \vdash A \multimap C, B \multimap D}}{(A \otimes B) \multimap (C \parr D) \vdash (A \multimap C) \parr (B \multimap D)}$$

is not a FILL proof as none of the applications of the implication right rule satisfy the FILL side-condition; C depends on B in the first application of the implication right rule and $B \multimap D$ depends on A in the second. This shows the similarity between $\parr$ and $\vee$ as the sequent $A \Rightarrow (C \vee D) \vdash (A \Rightarrow C) \vee (A \Rightarrow D)$ is provable in Classical Logic but not in Intuitionistic Logic.

3 Cut-Elimination

In this section we show that Classical Linear Logic and FILL satisfy the cut-elimination property. We actually give an algorithm transforming an arbitrary Classical Linear Logic proof into a cut-free proof with the same end-sequent such that the property of being a FILL proof is preserved. To this end we show that each step of the algorithm preserves the property of being a FILL proof, and moreover, each step decreases the sets of dependencies. This enables us to apply a step of the algorithm to a subproof above a valid FILL application of the implication right rule without risking to make it FILL-invalid. Towards the goal of proving this theorem we start with some preliminary results.

Some Preliminary Results

Our first preliminary results are two small technical lemmas needed to do the induction proofs later on.

Lemma 6. *Assume that we have a proof τ of $\Gamma, B \vdash \Delta, A$ such that neither of the formula occurrences A and B is introduced by the last rule used, and such that they both are inherited from an immediate subproof π of τ. Then A depends on B in τ iff A depends on B in the immediate subproof π.*

Lemma 7. *Assume that we have a proof τ of $\Gamma, !A, !A \vdash \Delta$ such that neither of the occurrences of the formula $!A$ is introduced by the last rule used, and such that they both are inherited from an immediate subproof π of τ. Then the proof with end-sequent $\Gamma, !A \vdash \Delta$ obtained by applying C_L to τ has the same dependencies as the proof with the same end-sequent obtained by applying C_L to π and then applying the last rule of τ.*

The previous lemma deals with contractions on the left hand side; there is a dual lemma dealing with contractions on the right hand side. The lemmas themselves are duals, but the proofs are not because of the asymmetry of the linear implication rules. The next lemma makes clear that the notion of dependency makes all rules for assignment of dependencies *monotonic* in the sense that if we take a proof

$$\dfrac{\begin{array}{c}\vdots\,\pi_i\\ \Gamma_i \vdash \Delta_i\end{array}}{\Gamma \vdash \Delta}$$

and replace each subproofs π_i by another proof ϖ_i with the same end-sequent as π_i such that ϖ_i has fewer dependencies than π_i, then the proof obtained

$$\dfrac{\begin{array}{c}\vdots\,\varpi_i\\ \Gamma_i \vdash \Delta_i\end{array}}{\Gamma \vdash \Delta}$$

has fewer dependencies than the original proof.

Lemma 8. *Assume that the proof τ is obtained from the proofs π_i by applying a proof-rule r, and that for each proof π_i there is a proof ϖ_i with the same end-sequent as π_i such that the end-sequent of ϖ_i has fewer dependencies than the end-sequent of π_i. Then the end-sequent of the proof σ obtained from the proofs ϖ_i by applying r has fewer dependencies than the end-sequent of τ. If τ is a FILL proof then each π_i is a FILL proof, and if furthermore each ϖ_i is a FILL proof, then σ is a FILL proof too.*

Our next lemma deals with the cases where a cut can be pushed upwards to the right hand side of a proof tree, that is, where the proof

$$\dfrac{\Gamma_1 \vdash A, \Delta_1 \qquad \dfrac{\Gamma_2, A \vdash \Delta_2}{\Gamma, A \vdash \Delta}\,r}{\Gamma, \Gamma_1 \vdash \Delta, \Delta_1}$$

is replaced by the proof

$$\dfrac{\dfrac{\Gamma_1 \vdash A, \Delta_1 \qquad \Gamma_2, A \vdash \Delta_2}{\Gamma_2, \Gamma_1 \vdash \Delta_2, \Delta_1}}{\Gamma, \Gamma_1 \vdash \Delta, \Delta_1}\, r$$

In these situations we have to worry, in particular, about the rules $!_R$ or $?_L$ since these are the only rules with conditions on the form of inactive formulae.

Lemma 9. *Let the proof τ of $\Gamma_2, \Gamma_1 \vdash \Delta_2, \Delta_1$ be obtained from the proof π_1 of $\Gamma_1 \vdash A, \Delta_1$ and the proof π_2 of $\Gamma_2, A \vdash \Delta_2$ by a cut. Assume that the last rule r of π_2 does not introduce the formula A. If r is an instance of $!_R$ or $?_L$ then assume furthermore that each formula in Γ_1 has a $!$ at the top level and each formula in Δ_1 has a $?$ at the top level. Let σ be the proof with conclusion $\Gamma_2, \Gamma_1 \vdash \Delta_2, \Delta_1$ obtained by applying r to the proof obtained by applying the cut rule to π_1 and the immediate subproof of π_2 from which A is inherited. Then*

- *the conclusion of σ has the same dependencies as the conclusion of τ;*
- *if τ is a proof in FILL then σ is a proof in FILL too.*

The previous lemma deals with pushing a cut up to the right hand side of a proof tree; there is a dual lemma dealing with pushing a cut up to the left hand side of a tree. These lemmas are dual, but their proofs are not because of the asymmetry of the linear implication rules.

The Key-Cases

As is usual in cut-elimination proofs we need to define the degree of a formula.

Definition 10. The degree $\partial(A)$ of a formula A is defined inductively as follows:

1. $\partial(A) = 1$ when A is atomic
2. $\partial(A \otimes B) = \partial(A ⅋ B) = \partial(A \multimap B) = max\{\partial(A), \partial(B)\} + 1$
3. $\partial(!A) = \partial(?A) = \partial(A) + 1$

The degree of a cut is defined to be the degree of the formula cut. The degree of a proof is defined as the supremum of the degrees of the cuts in the proof.

Our next lemma deals with all the key-cases. The $(\otimes_R, \otimes_L)$, (I_R, I_L), $(\multimap_R, \multimap_L)$, $(!_R, !_L)$ and $(!_R, W_L)$ key-cases can be found in Figure 3. We also have $(⅋_R, ⅋_L)$, $(\bot_R, \bot_L)$, $(?_R, ?_L)$ and $(W_R, ?_L)$ key-cases, but they are omitted as they are dual to the $(\otimes_R, \otimes_L)$, (I_R, I_L), $(!_R, !_L)$ and $(!_R, W_L)$ cases, respectively.

Lemma 11. *Let the proof τ be an instance of one of the key-cases. Assume that the immediate subproofs of τ have degrees strictly less than the degree of τ. Let σ be the proof obtained by changing τ as prescribed. Then σ has degree strictly less than the degree of τ and moreover*

- *if τ is a proof in FILL then the end-sequent of σ has fewer dependencies than the end-sequent of τ;*
- *if τ is a proof in FILL then σ is a proof in FILL too.*

Note in Lemma 11 that the end-sequent of σ has fewer dependencies than the end-sequent of τ *under the assumption that* τ is a proof in FILL.

Fig. 3. The Key-Cases

$$\dfrac{\dfrac{\overset{\vdots\,\pi_1}{\Gamma_1 \vdash A, \Delta_1} \quad \overset{\vdots\,\pi_2}{\Gamma_2 \vdash B, \Delta_2}}{\Gamma_1, \Gamma_2 \vdash A \otimes B, \Delta_1, \Delta_2} \quad \dfrac{\overset{\vdots\,\pi_3}{\Gamma_3, A, B \vdash \Delta_3}}{\Gamma_3, A \otimes B \vdash \Delta_3}}{\Gamma_3, \Gamma_1, \Gamma_2 \vdash \Delta_3, \Delta_1, \Delta_2} \rightsquigarrow \dfrac{\overset{\vdots\,\pi_1}{\Gamma_1 \vdash A, \Delta_1} \quad \dfrac{\overset{\vdots\,\pi_2}{\Gamma_2 \vdash B, \Delta_2} \quad \overset{\vdots\,\pi_3}{\Gamma_3, A, B \vdash \Delta_3}}{\Gamma_3, A, \Gamma_2 \vdash \Delta_3, \Delta_2}}{\Gamma_3, \Gamma_1, \Gamma_2 \vdash \Delta_3, \Delta_2, \Delta_1}$$

$$\dfrac{\dfrac{}{\vdash I} \quad \dfrac{\overset{\vdots\,\pi}{\Gamma \vdash \Delta}}{\Gamma, I \vdash \Delta}}{\Gamma \vdash \Delta} \rightsquigarrow \overset{\vdots\,\pi}{\Gamma \vdash \Delta}$$

$$\dfrac{\dfrac{\overset{\vdots\,\pi_1}{\Gamma_1, A \vdash B, \Delta_1}}{\Gamma_1 \vdash A \multimap B, \Delta_1} \quad \dfrac{\overset{\vdots\,\pi_2}{\Gamma_2 \vdash A, \Delta_2} \quad \overset{\vdots\,\pi_3}{\Gamma_3, B \vdash \Delta_3}}{\Gamma_2, \Gamma_3, A \multimap B \vdash \Delta_3, \Delta_2}}{\Gamma_2, \Gamma_3, \Gamma_1 \vdash \Delta_3, \Delta_2, \Delta_1} \rightsquigarrow \dfrac{\overset{\vdots\,\pi_2}{\Gamma_2 \vdash A, \Delta_2} \quad \dfrac{\overset{\vdots\,\pi_1}{\Gamma_1, A \vdash B, \Delta_1} \quad \overset{\vdots\,\pi_3}{\Gamma_3, B \vdash \Delta_3}}{\Gamma_3, \Gamma_1, A \vdash \Delta_3, \Delta_1}}{\Gamma_3, \Gamma_1, \Gamma_2 \vdash \Delta_3, \Delta_1, \Delta_2}$$

$$\dfrac{\dfrac{\overset{\vdots\,\pi_1}{!\Gamma_1 \vdash A, ?\Delta_1}}{!\Gamma_1 \vdash !A, ?\Delta_1} \quad \dfrac{\overset{\vdots\,\pi_2}{\Gamma_2, A \vdash \Delta_2}}{\Gamma_2, !A \vdash \Delta_2}}{\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1} \rightsquigarrow \dfrac{\overset{\vdots\,\pi_1}{!\Gamma_1 \vdash A, ?\Delta_1} \quad \overset{\vdots\,\pi_2}{\Gamma_2, A \vdash \Delta_2}}{\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1}$$

$$\dfrac{\dfrac{\overset{\vdots\,\pi_1}{!\Gamma_1 \vdash A, ?\Delta_1}}{!\Gamma_1 \vdash !A, ?\Delta_1} \quad \dfrac{\overset{\vdots\,\pi_2}{\Gamma_2 \vdash \Delta_2}}{\Gamma_2, !A \vdash \Delta_2}}{\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1} \rightsquigarrow \begin{array}{c} \overset{\vdots\,\pi_2}{\Gamma_2 \vdash \Delta_2} \\ \hline\hline \Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1 \end{array}$$

Fig. 4. The Pseudo Key-Case

$$\dfrac{\dfrac{\overset{\vdots\,\pi_1}{!\Gamma_1 \vdash A, ?\Delta_1}}{!\Gamma_1 \vdash !A, ?\Delta_1} \quad \dfrac{\overset{\vdots\,\pi_2}{\Gamma_2, !A, !A \vdash \Delta_2}}{\Gamma_2, !A \vdash \Delta_2}}{\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1} \rightsquigarrow \begin{array}{c} \dfrac{\dfrac{\overset{\vdots\,\pi_1}{!\Gamma_1 \vdash A, ?\Delta_1}}{!\Gamma_1 \vdash !A, ?\Delta_1} \quad \dfrac{\dfrac{\overset{\vdots\,\pi_1}{!\Gamma_1 \vdash A, ?\Delta_1}}{!\Gamma_1 \vdash !A, ?\Delta_1} \quad \overset{\vdots\,\pi_2}{\Gamma_2, !A, !A \vdash \Delta_2}}{\Gamma_2, !\Gamma_1, !A \vdash \Delta_2, ?\Delta_1}}{\Gamma_2, !\Gamma, !\Gamma_1 \vdash \Delta_2, ?\Delta_1, ?\Delta_1} \\ \hline\hline \Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1 \end{array}$$

The Pseudo Key-Cases

Now we deal with what we call the pseudo key-cases, which differ from the key-cases in that they do not decrease the degree of the proof involved. The $(!_R, C_L)$ key-case can be found in Figure 4. We also have a $(C_R, ?_L)$ key-case, but it is omitted as it is dual to the $(!_R, C_L)$ case. The following lemma says that the pseudo key-cases behave properly with respect to dependencies. But the hard work is done in Lemma 14, where we show how to perform additional modifications when applying the pseudo key-cases such that the degree of the involved proof *does* decrease.

Lemma 12. *Let the proof τ be an instance of one of the pseudo key-cases. Let σ be the proof obtained by changing τ as prescribed. Then*

- *the end-sequent of σ has the same dependencies as the end-sequent of τ;*
- *if τ is a proof in FILL then σ is a proof in FILL too.*

Let us examine the $(!_R, C_L)$ pseudo key-case, which does not decrease the degree of the involved proof, as mentioned above. Thus instead of just modifying the proof as prescribed, we have to "track" the cut formula $!A$ upwards in the right hand side immediate subproof which has C_L as the last rule. When we follow an antecedent $!A$ formula upwards in a proof, then a track ends due to one of the following reasons: The formula is introduced by an instance of the $!_L$ rule, the formula is absorbed by an instance of the W_L rule, or finally, we end up in an axiom. Note that on the way upwards the formula $!A$ might have proliferated into several copies while passing instances of the C_L rule. In each of the mentioned end-points there is an appropriate way to modify the proof such that $!A$ is replaced by $!\Gamma$ and $?\Delta$ is added on the succedent in a way that decreases the degree. For example, the following proof

$$\frac{\dfrac{!\Gamma \vdash A, ?\Delta}{!\Gamma \vdash !A, ?\Delta}\,!_R \qquad \dfrac{\dfrac{\begin{array}{c}\overline{!A \vdash !A}\\ \vdots \\ \Gamma_1, !A \vdash \Delta_1\end{array} \qquad \begin{array}{c}\overline{!A \vdash !A}\\ \vdots \\ \Gamma_2, !A \vdash \Delta_2\end{array}}{\Gamma', !A, !A \vdash \Delta'}}{\Gamma', !A \vdash \Delta'}}{\Gamma', !\Gamma \vdash \Delta', ?\Delta}$$

where none of the two tracks are proliferated by passing instances of the C_L rule and they both both end up in an axiom, is modified as follows

$$\dfrac{\begin{array}{c}\dfrac{!\Gamma \vdash A, ?\Delta}{!\Gamma \vdash !A, ?\Delta}\,!_R \\ \vdots \\ \Gamma_1, !\Gamma \vdash \Delta_1, ?\Delta\end{array} \qquad \begin{array}{c}\dfrac{!\Gamma \vdash A, ?\Delta}{!\Gamma \vdash !A, ?\Delta}\,!_R \\ \vdots \\ \Gamma_2, !\Gamma \vdash \Delta_2, ?\Delta\end{array}}{\dfrac{\Gamma', !\Gamma, !\Gamma \vdash \Delta', ?\Delta, ?\Delta}{\Gamma', !\Gamma \vdash \Delta', ?\Delta}}$$

Note that a $(!_R, C_L)$ pseudo key-case was performed in the process of following the track(s) upwards. This is formalised in the next lemma which takes care of every situation where we have a cut such that the last rule of the left hand side immediate subproof is an instance of the $!_R$ rule. The lemma is proved by induction using the notion of height of a proof:

Definition 13. The height $h(\tau)$ of a proof τ is defined inductively as follows:

1. if τ is an instance of a zero-premiss rule, then $h(\tau) = 1$,
2. if τ is obtained from the proof π by applying an instance of a one-premiss rule, then $h(\tau) = h(\pi) + 1$,
3. if τ is obtained from the proofs π_1 and π_2 by applying an instance of a two-premiss rule, then $h(\tau) = max\{h(\pi_1), h(\pi_2)\} + 1$.

Now the promised lemma.

Lemma 14. *Assume $!A^n$ denotes a list of n occurrences of the formula $!A$, and let a proof τ be given as follows*

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\pi_1 \\ !\Gamma_1 \vdash A, ?\Delta_1\end{array}}{!\Gamma_1 \vdash !A, ?\Delta_1} \qquad \dfrac{\begin{array}{c}\vdots\,\pi_2 \\ \Gamma_2, !A^n \vdash \Delta_2\end{array}}{\Gamma_2, !A \vdash \Delta_2}}{\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1}$$

where the subproofs π_1 and π_2 have degrees strictly less than the degree of τ. Then we can construct a proof σ of $\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1$ such that σ has degree strictly less than the degree of τ and moreover

- *the end-sequent of σ has fewer dependencies than the end-sequent of τ;*
- *if τ is a proof in FILL then σ is a proof in FILL too.*

Proof. The proof is by induction on $h(\pi_2)$. In what follows, the last rule used in π_2 is denoted by r and we proceed in a case by case basis.

1. If r is an instance of an axiom. We use the proof obtained by applying $!_R$ to π_1.

2. If one of the n occurrences of the formula $!A$ in the end-sequent of π_2 is introduced by r which is an instance of $!_L$. We have two subcases. If $n = 1$ then we have a $(!_R, !_L)$ key-case, so we change τ according to the table of key-cases in Figure 3 cf. Lemma 11. If $n > 1$ then we have the following situation

$$
\begin{prooftree}
\AxiomC{$!\Gamma_1 \vdash A, ?\Delta_1$}
\UnaryInfC{$!\Gamma_1 \vdash !A, ?\Delta_1$}
\AxiomC{$\Gamma_2, !A^{n-1}, A \vdash \Delta_2$}
\UnaryInfC{$\Gamma_2, !A^{n-1}, !A \vdash \Delta_2$}
\doubleLine
\UnaryInfC{$\Gamma_2, !A \vdash \Delta_2$}
\BinaryInfC{$\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1$}
\end{prooftree}
$$

where we first change the proof in accordance with Lemma 7 to obtain the $(!_R, C_L)$ pseudo key-case

$$
\begin{prooftree}
\AxiomC{$!\Gamma_1 \vdash A, ?\Delta_1$}
\UnaryInfC{$!\Gamma_1 \vdash !A, ?\Delta_1$}
\AxiomC{$\Gamma_2, !A^{n-1}, A \vdash \Delta_2$}
\doubleLine
\UnaryInfC{$\Gamma_2, !A, A \vdash \Delta_2$}
\UnaryInfC{$\Gamma_2, !A, !A \vdash \Delta_2$}
\UnaryInfC{$\Gamma_2, !A \vdash \Delta_2$}
\BinaryInfC{$\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1$}
\end{prooftree}
$$

which we change as prescribed in Figure 4 cf. Lemma 12 to

$$
\begin{prooftree}
\AxiomC{$!\Gamma_1 \vdash A, ?\Delta_1$}
\UnaryInfC{$!\Gamma_1 \vdash !A, ?\Delta_1$}
\AxiomC{$!\Gamma_1 \vdash A, ?\Delta_1$}
\UnaryInfC{$!\Gamma_1 \vdash !A, ?\Delta_1$}
\AxiomC{$\Gamma_2, !A^{n-1}, A \vdash \Delta_2$}
\doubleLine
\UnaryInfC{$\Gamma_2, !A, A \vdash \Delta_2$}
\UnaryInfC{$\Gamma_2, !A, !A \vdash \Delta_2$}
\BinaryInfC{$\Gamma_2, !\Gamma_1, !A \vdash \Delta_2, ?\Delta_1$}
\BinaryInfC{$\Gamma_2, !\Gamma_1, !\Gamma_1 \vdash \Delta_2, ?\Delta_1, ?\Delta_1$}
\doubleLine
\UnaryInfC{$\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1$}
\end{prooftree}
$$

on which we apply Lemma 9 to obtain

$$
\begin{prooftree}
\AxiomC{$!\Gamma_1 \vdash A, ?\Delta_1$}
\UnaryInfC{$!\Gamma_1 \vdash !A, ?\Delta_1$}
\AxiomC{$!\Gamma_1 \vdash A, ?\Delta_1$}
\UnaryInfC{$!\Gamma_1 \vdash !A, ?\Delta_1$}
\AxiomC{$\Gamma_2, !A^{n-1}, A \vdash \Delta_2$}
\doubleLine
\UnaryInfC{$\Gamma_2, !A, A \vdash \Delta_2$}
\BinaryInfC{$\Gamma_2, !\Gamma_1, A \vdash \Delta_2, ?\Delta_1$}
\UnaryInfC{$\Gamma_2, !\Gamma_1, !A \vdash \Delta_2, ?\Delta_1$}
\BinaryInfC{$\Gamma_2, !\Gamma_1, !\Gamma_1 \vdash \Delta_2, ?\Delta_1, ?\Delta_1$}
\doubleLine
\UnaryInfC{$\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1$}
\end{prooftree}
$$

which contains a $(!_R, !_L)$ key-case which we eliminate as prescribed in Figure 3 cf. Lemma 11 and get a proof

$$\dfrac{\dfrac{!\Gamma_1 \vdash A, ?\Delta_1 \qquad \dfrac{\dfrac{!\Gamma_1 \vdash A, ?\Delta_1}{!\Gamma_1 \vdash !A, ?\Delta_1} \qquad \dfrac{\Gamma_2, !A^{n-1}, A \vdash \Delta_2}{\Gamma_2, !A, A \vdash \Delta_2}}{\Gamma_2, !\Gamma_1, A \vdash \Delta_2, ?\Delta_1}}{\Gamma_2, !\Gamma_1, !\Gamma_1 \vdash \Delta_2, ?\Delta_1, ?\Delta_1}}{\Gamma_2, !\Gamma_1 \vdash \Delta_2, ?\Delta_1}$$

on which we apply the induction hypothesis on the appropriate subproof.

3. If one of the n occurrences of the formula $!A$ in the end-sequent of π_2 is introduced by r which is an instance of W_L. This situation is analogous to the previous case.
4. If one of the n occurrences of the formula $!A$ in the end-sequent of π_2 is introduced by r which is an instance of C_L. We apply the induction hypothesis to the immediate subproof of π_2 where we consider the appropriate $n+1$ occurrences of $!A$.
5. If none of the n occurrences of the formula $!A$ in the end-sequent of π_2 is introduced by r. There are two subcases depending upon whether or not all of the n occurrences of the formula $!A$ in the end-sequent of π_2 are inherited from the same immediate subproof. They are both dealt with in a way similar to case 2 and 3.

□

There is a dual to Lemma 14; and it is the case that the proofs are also dual.

Putting the Proof Together

The following lemma is the engine of the cut-elimination proof.

Lemma 15. *Let the proof τ of $\Gamma_2, \Gamma_1 \vdash \Delta_2, \Delta_1$ be obtained from the proof π_1 of $\Gamma_1 \vdash A, \Delta_1$ and the proof π_2 of $\Gamma_2, A \vdash \Delta_2$ by a cut. Assume that π_1 and π_2 have degrees strictly less than the degree of τ. Then we can construct a proof σ of $\Gamma_2, \Gamma_1 \vdash \Delta_2, \Delta_1$ such that the proof σ has degree strictly less than the degree of τ and moreover*

- *if τ is a proof in FILL then the end-sequent of σ has fewer dependencies than the end-sequent of τ;*
- *if τ is a proof in FILL then σ is a proof in FILL too.*

Proof. Induction on $h(\pi_1) + h(\pi_2)$. In what follows, the last rules in π_1 and π_2 are denoted by r_1 and r_2, respectively. The formula A is denoted the principal formula. We proceed case by case, dual cases are omitted.

1. If r_1 is an instance of an axiom. We use π_2.
2. If r_1 is an instance of $!_R$ that introduces the principal formula. We use Lemma 14.

3. If r_1 is an instance of $!_R$ that does not introduce the principal formula. Thus, $A = ?C$ for some C, and π_1 looks as follows:

$$\frac{!\Gamma_1 \vdash B, ?C, ?\Delta_1}{!\Gamma_1 \vdash !B, ?C, ?\Delta_1}$$

If the principal formula $?C$ is not introduced by r_2, then we change τ according to Lemma 9 (note that r_2 can not be an instance of $!_R$ or $?_L$) and use the induction hypothesis. If the principal formula $?C$ is introduced by r_2, that is, π_2 looks as follows:

$$\frac{!\Gamma_2, C \vdash ?\Delta_2}{!\Gamma_2, ?C \vdash ?\Delta_2}$$

then we change τ according to the dual to Lemma 9 (note that each formula in $!\Gamma_2$ has a ! at the top level and each formula in $?\Delta_2$ has a ? at the top level) and use the induction hypothesis.
4. If r_1 is an instance of $?_L$. Similar to case 3.
5. If r_1 does not introduce the principal formula. We change τ according to the dual to Lemma 9 and use the induction hypothesis.
6. If r_1 introduces the principal formula on the right hand side and r_2 introduces the principal formula on the left hand side. We have a $(\otimes_R, \otimes_L)$, (I_R, I_L), $(⅋_R, ⅋_L)$, $(\bot_R, \bot_L)$ or $(\multimap_R, \multimap_L)$ key-case. We change τ as prescribed cf. Lemma 11.

□

Lemma 16. *Given a proof τ of $\Gamma \vdash \Delta$, we can construct a proof σ of the same sequent with strictly lower degree than the degree of τ such that*

- *if τ is a proof in FILL then the end-sequent of σ has fewer dependencies than the end-sequent of τ;*
- *if τ is a proof in FILL then σ is a proof in FILL too.*

Proof. Induction on $h(\tau)$. If the last rule used in τ is not a cut with the same degree as the degree of τ, we are done by using the induction hypothesis on the immediate subproofs of τ. If the last rule in τ is a cut with the same degree as the degree of τ, we apply the induction hypothesis on the immediate subproofs of τ, and obtain a proof π which looks as follows

$$\frac{\begin{array}{c}\vdots\, \pi_1 \\ \Gamma_1 \vdash A, \Delta_1\end{array} \qquad \begin{array}{c}\vdots\, \pi_2 \\ \Gamma_2, A \vdash \Delta_2\end{array}}{\Gamma_2, \Gamma_1 \vdash \Delta_2, \Delta_1}$$

where the proofs π_1 and π_2 have strictly lower degree than the degree of π. We then use Lemma 15. □

And now the Hauptsatz.

Theorem 17. *Given a proof τ of $\Gamma \vdash \Delta$, we can construct a cut-free proof σ of the same sequent such that*

- *if τ is a proof in FILL then the end-sequent of σ has fewer dependencies than the end-sequent of τ;*
- *if τ is a proof in FILL then σ is a proof in FILL too.*

Proof. Iteration of Lemma 16. □

Note in Theorem 17 that the end-sequent of σ has fewer dependencies than the end-sequent of τ *under the assumption that* τ is a proof in FILL. This assumption is inherited from Lemma 11 via Lemma 15 and Lemma 16.

4 Conclusions

We have described a cut-elimination procedure for Classical Linear Logic that works for FILL, an intuitionistic and more restricted system. In [HdP93] a term assignment system was proposed to handle the "dependency condition" of our Definition 4. There was a small mistake in that paper which is corrected here. The dependency-relation introduced in this paper enables the side-condition of the implication right rule to be stated in a way that captures the underlying notion of intuitionistic implication, and moreover, that enables the proof of the cut-elimination theorem to go through in a straightforward manner. We remark that our formulation using dependencies might please traditional logicians, who may not want a term assignment decorating their usual sequent calculus proofs, but who are fairly used to side conditions on their rules. This characteristic feature of the formal system seems of independent interest.

A similar approach to ours was pursued by de Paiva and Pereira [dPP95] for Intuitionistic Logic, as opposed to Intuitionistic Linear Logic. Their approach is more akin to a primitive kind of term assignment, as their sequents are of the form

$$A_1(n_1), \ldots, A_k(n_k) \Rightarrow B_1/S_1, \ldots, B_m/S_m$$

where the n_i are natural numbers and the S's are sets of natural numbers that code up the dependency relation.

One may ask whether there is a connection between FILL and the linear version of Parigot's $\lambda\mu$-calculus, [Par92], put forward by Bierman, [Bie97]. Our immediate answer to this question is negative - the concern of the $\lambda\mu$-calculus is to control the process of classical cut-elimination whereas the point of FILL is that it allows for multible conclusions in an intuitionistic framework. Another way of putting the orthogonality of the approaches is to say that the $\lambda\mu$-calculus is about interleaving (and intertwining) intuitionistic computations so much as they stop being intuitionistic whereas FILL is about keeping intuitionistic computations as parallel as possible so that the whole computation can still be considered intuitionistic.

Recently, linear functional programming, which is based on Linear Logic rather than Intuitionistic Logic, has been proposed as a means of adding resource control into functional programming. All the work done so far on linear functional programming is based on Intuitionistic Linear Logic, the fragment considered by Lafont and Girard in [GL87]. For this fragment, since derivations have a single conclusion, it is easier to formulate a Natural Deduction version, and hence the similarity with standard functional programming is clearer. To extend this kind of approach to the system of FILL seems very promising. Because FILL is a multiple conclusion system this logic should give rise to an intrinsically parallel version of linear functional programming and as such it will enhance resource control in a parallel-based environment.

Acknowledgements: We would like to thank Gianluigi Bellin, Gavin Bierman, Martin Hyland and Luiz Carlos Pereira for stimulating discussions on the subject of this paper. We also thank Mathias Kegelmann for reading a draft of this work and providing many helpful comments and suggestions. We have used Paul Taylor's macros to produce the proof-rules.

References

[BdP96] T. Braüner and V. de Paiva. Cut-elimination for full intuitionistic linear logic. Technical Report 395, Computer Laboratory, University of Cambridge, 1996. 27 pages. Also available as Technical Report RS-96-10, BRICS, University of Aarhus.

[Bel96] G. Bellin. Subnets of proofnets in multiplicative linear logic with MIX. Manuscript, 1996.

[Bie96] G. Bierman. A note on full intuitionistic linear logic. *Annals of Pure and Applied Logic*, 79, 1996.

[Bie97] G. Bierman. A classical linear λ-calculus. 1997. Submitted for publication.

[CS96] J. R. B. Cockett and R. A. G. Seely. Proof theory for full intuitionistic linear logic, bilinear logic, and MIX categories. Manuscript, 1996.

[dP88] V. C. V. de Paiva. *The Dialectica Categories.* PhD thesis, Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, 1988. Published as Computer Laboratory Technical Report 213.

[dPP95] V. de Paiva and L. C. Pereira. A new proof system for intuitionistic logic. *The Bulletin of Symbolic Logic*, 1:101, 1995.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50, 1987.

[Gir94] J.-Y. Girard. Light linear logic. Manuscript, 1994.

[GL87] J.-Y. Girard and Y. Lafont. Linear logic and lazy computation. In *Proceedings of TAPSOFT '87, LNCS*, volume 250. Springer-Verlag, 1987.

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge University Press, 1989.

[HdP93] M. Hyland and V. de Paiva. Full intuitionistic linear logic (extended abstract). *Annals of Pure and Applied Logic*, 64, 1993.

[Par92] M. Parigot. $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction. In *Proceedings of LPAR, LNCS*, volume 624. Springer-Verlag, 1992.

[Sch91] H. Schellinx. Some syntactic observations on linear logic. *Journal of Logic and Computation*, 1, 1991.

[Shi94] M. Shirahata. *Linear Set Theory.* PhD thesis, Stanford University, 1994.