The Uniform Proof-Theoretic Foundation of Linear Logic Programming

by

James Harland and David Pym

The Uniform Proof-Theoretic Foundation
of Linear Logic Programming

LFCS Report Series **ECS-LFCS-90-124**

LFCS
Department of Computer Science
University of Edinburgh
The King's Buildings
Edinburgh EH9 3JZ

November 1990

THE UNIFORM PROOF–THEORETIC FOUNDATION OF LINEAR LOGIC PROGRAMMING [1] [2]

James Harland
University of Melbourne
Australia
jah@au.oz.mu.cs

David Pym
University of Edinburgh
Scotland, U.K.
dpym@uk.ac.ed.lfcs

Abstract

We present a theory of *logic programming* for Girard's *linear logic.* In the spirit of recent work of Miller *et al.,* we identify suitable classes of *definite formulae* and *goal formulae* by considering classes of formulae that resemble the Harrop formulae of intuitionistic first-order logic. We isolate the appropriate notion of *uniform proof* in linear logic, and show that this characterizes *resolution* proofs. Resolution proofs in linear logic are somewhat difficult to define. This difficulty arises from the need to decompose definite formulae into a form suitable for the use of the linear resolution rule, a rule which requires the clause selected to be *deleted* after use. Further difficulties are encountered when we consider the addition of the *modality* ! (of course) to our fragment of linear logic.

We provide an elementary *quantale semantics* for our logic programs and give an appropriate *completeness theorem.*

We consider a *translation* — resembling those of Girard — of the intuitionistic hereditary Harrop formulae and intuitionistic uniform proofs into our framework, and show that certain properties are preserved under this translation.

We sketch the design of an *interpreter* for linear logic programs.

Keywords: Definite Formulae; Linear Logic; Logic Programming; Modules; Proof Theory; Resolution; Quantales.

1 Introduction

An interesting recent development in logic of some significance for theoretical computer science is *linear logic* [14], [15], a logic designed with bounded resources in mind. One of the distinguishing features of this logic is that the two sequents $\phi \vdash \psi$ and $\phi, \phi \vdash \psi$ may behave very differently, as in the former sequent the formula ϕ may be used once only, whereas in the second it may be used twice. Thus there is a difference between using a formula *once* in a proof and using it *many* times in the same proof. Linear logic has been applied to the study of concurrency, proving to be particularly valuable for reasoning

[1]Harland is supported by a grant of the Australian Research Council through the Machine Intelligence Project. Pym is supported by U.K. Science and Engineering Research Council grant GR/E 78487, "Computer Assisted Formal Reasoning: Logical Frameworks"; and by European Community ESPRIT Basic Research Action 3245, "Logical Frameworks: Design, Implementation and Experiment".

[2]Correspondence to: David Pym, Laboratory for Foundations of Computer Science, University of Edinburgh, The King's Buildings, Edinburgh EH9 3JZ, Scotland, U.K..

about *Petri nets* [13], [8]. Recently, there has been some work towards object-oriented logic programming which uses certain fragments of linear logic [2].

In this paper we present a proof-theoretic analysis of how a certain fragment of linear logic can be used as a logic programming language.

The starting point for a theory of logic programming in classical Horn clause logic is the cut-free sequent calculus LK [12], [19]. Similarly, the sequent calculus LJ [12] provides the basis for logic programming with the *hereditary Harrop* fragment of intuitionistic logic [20], [21]. A sequent calculus presentation of linear logic can be found in [14], [15], [4], and in Appendix A. Linear sequents are expressions of the form

$$\phi_1, \ldots, \phi_m \vdash \psi_1, \ldots, \psi_n \,,$$

and we shall let Γ and Δ range over antecedents and succedents, respectively.

Following the work of Miller *et al.* for intuitionistic logic [20], [21], [16], our analysis rests on the identification of certain classes of linear logic formulae.[3]

The first and most crucial class is that of *definite formulae.* These are the formulae that can occur as the components of an antecedent of a linear sequent, *i.e.,* these are the components of a *linear logic program.* In the case of intuitionistic logic, definite formulae are, essentially, the conjunctive and implicational fragment of the logic. Correspondingly in linear logic, we take definite formulae to consist of the *multiplicative conjunctive* fragment $\otimes$, the *additive conjunctive* fragment $\&$, and the implicational fragment $-\!\circ$: more precisely, linear definite formulae are given by the grammar

$$D \ ::= \ L \mid D_1 \otimes D_2 \mid D_1 \,\&\, D_2 \mid G -\!\circ L \mid \bigwedge x \,.\, D \,,$$

where L denotes the class of *positive literals* — the constants of linear logic and atomic formulae — and G denotes the class of *goal formulae*, to be discussed below. From a programming point of view, definite formulae can be considered to those formulae that are *deterministic* in their formation.[4]

In order to complete our description of the language of linear logic programming we must describe the class of goal formulae *goal formulae.* These are the formulae that can occur as the components of a succedent of a linear sequent. Goal formulae are given by the grammar

$$\begin{aligned} G \ ::= \ & L \mid L^{\perp} \mid G \otimes G \mid G \oplus G \mid G \maltese G \mid G \,\&\, G \\ & \mid D -\!\circ G \mid \bigwedge x \,.\, G \mid \bigvee x \,.\, G \,. \end{aligned}$$

Note that negative literals, *i.e.,* those of the form $L^{\perp}$, may only appear in goals. The reason for this restriction will be discussed in Section 3.

[3]Miller *et al.* identify the class of *hereditary Harrop formulae* as determining a suitable fragment of (minimal and) intuitionistic logic for use as a logic programming language.

[4]In this respect it is not clear whether or not we should include & in the class of definite formulae, for although it is a conjunction, its left rule has rather disjunctive behaviour (see Appendix A). We choose to include it because it satisfies the required proof-theoretic properties, even though its inclusion is philosophically dubious.

We can extend the classes of definite formulae and goal formulae with certain instances of the *exponentials* ! (of course) and ? (why not), but we delay consideration of this point until we consider the translation of intuitionistic logic into linear logic in Section 6.

What, then, is to be our notion of *computational proof* or *operational proof*? We shall develop such a notion in two steps.

The first step, the more proof-theoretic part, deals with the right rules and their *permutation* properties with respect to the left rules.[5]

Suppose that we are faced with a sequent — a goal — of the form

$$\phi_1, \ldots, \phi_m \vdash \psi_1, \ldots, \psi_n$$

and that we wish to attempt to construct a proof of this sequent by considering the rules of linear sequent calculus as *reduction operators* from conclusion to premisses.[6]

Our approach is in the spirit of one of the distinguishing features of logic programming, namely *goal-directed* proof-search. More precisely, there is a search operation corresponding to each logical connective, and when searching for a proof of a given goal one applies the search operation that corresponds to the outermost connective of that goal, and then to the outermost connective of each subgoal so generated, *etc.*. For (classical) Horn clause goals [19], *i.e.,* existentially quantified conjunctions of atoms, this corresponds to using unification to "delay" the choice of witnesses for the variables, and a strategy for selecting a particular atom as the next subgoal. For a larger language, such as first-order hereditary Harrop formulae [21], we need a richer set of primitive search operations, or *search primitives*, so that each distinct connective corresponds to a distinct search primitive.

This approach can be adopted in linear logic. For example, suppose that our sequent is of the form

$$\phi_1, \ldots, \phi_m \vdash \eta \mathbin{\&} \theta, \psi_1, \ldots, \psi_n \,;$$

the & -R rule, as presented in Appendix A,

$$\frac{\Gamma \vdash \phi, \Delta \qquad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \mathbin{\&} \psi, \Delta} \;\& \text{-R}$$

directs us to reduce to the subgoals

$$\phi_1, \ldots, \phi_m \vdash \eta, \psi_1, \ldots, \psi_n \qquad \text{and} \qquad \phi_1, \ldots, \phi_m \vdash \theta, \psi_1, \ldots, \psi_n \,.$$

However, the $\otimes$-R rule,

$$\frac{\Gamma \vdash \phi, \Delta \qquad \Gamma' \vdash \psi, \Delta'}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'} \;\otimes\text{-R},$$

is more complicated when considered as a reduction operator, in that it requires both the antecedent and the succedent to be *non-deterministically* split. The management of

[5]The reader who is interested in *Permutation Theorems* more generally should compare this work with that of Curry [10] and Kleene [18] for classical and intuitionistic logic, and Pym and Wallen [23] for the $\lambda\Pi$-calculus. Permutation theorems provide the key to understanding characterizations of proofs in sequent calculus such as *matrices* [7], [3], [24] and *proof nets* [14].

[6]Kleene [18] explains this in the case of the classical predicate calculus.

this non-determinism is a distinctive problem in the definition of an interpreter for linear logic programs.[7]

Thus we identify as our notion of ***uniform proof*** those proofs which when considered to be constructed from the *endsequent* or *root*, using the rules of linear sequent calculus as reduction operators, always use a right rule whenever such a rule is applicable. We prove that provided we restrict our sequents to be of the form

$$D_1, \ldots, D_m \vdash G_1, \ldots, G_n$$

such proofs completely characterize proofs in linear sequent calculus.[8] Such antecedents and succedents will be called *programs*, denoted by $\mathcal{P}$, and *goals*, denoted by $\mathcal{G}$, respectively.

The second step in our identification of a notion of *computational proof* for linear logic concerns the left rules and the representation of antecedents in such a way that a form of *resolution* completely characterizes uniform proofs.

In first-order Horn clauses and hereditary Harrop formulae, as presented in [20], the resolution rule can be considered to replace the $\wedge$-L, $\supset$-L and $\forall$-L rules. This is affected by the use of a decompostion $[\mathcal{P}]$ of a program $\mathcal{P}$ into a collection of *clauses* of the form $G \supset L$, where G is goal formula and L is an atomic formula.

In linear logic the resolution rule must replace the $\&$-L, $\otimes$-L, $\multimap$-L and $\bigwedge$-L rules. In fact, the resolution rule, which is to be considered as a reduction operator, is given by

$$\frac{\mathcal{D}[\vec{t}/\vec{x}] \vdash_R G, \mathcal{G} \qquad \{L\} \vdash_R L}{\mathcal{D} \cup \{G' \multimap L'\} \vdash_R L, \mathcal{G}}$$

where $\mathcal{D}$ denotes a set of definite formulae and $G \multimap L \in \bigcup_{\vec{t} \in \mathcal{U}}\{G'[\vec{t}/\vec{x}] \multimap L'[\vec{t}/\vec{x}]\}$, where $\mathcal{U}$ denotes the universe of terms. It is important to note that the calculated substitution $[\vec{t}/\vec{x}]$ is not restricted to the the left hand premiss of this rule: it applies to the whole of the derivation tree that lies above the same premiss of the last $\&$-R rule.[9] Furthermore, note that the absence of the structural rule of contraction in linear logic means that the clause $G' \multimap L'$ can not be retained in the premisses for further use, in contrast to the situation in classical and intuitionistic logic.

How are we to obtain a completeness theorem for resolution ? We first decompose a program into *clauses*, formulae of the form $G \multimap L$ or the form L. Let $[\mathcal{P}]$ denote this decomposition of the program $\mathcal{P}$. A *resolution proof* of the sequent $\mathcal{P} \vdash \mathcal{G}$ is now constructed as follows:

- Decompose the program $\mathcal{P}$ so that we consider sequents of the form $[\mathcal{P}] \vdash \mathcal{G}$;
- Use right rules to decompose the components of the goal until they are atomic;

[7]This point is discussed further in Section 7.

[8]Actually, this does not quite work: from the proof-theoretic point of view we need to consider a more *local* form of uniform proof in order to retain completeness. This point represents a significant additional difficulty with linear logic over intuitionistic logic and is discussed in some detail in Section 3.

[9]This point is discussed further in Sections 4 and 6.

- Use the resolution rule to invoke the clauses of $[\mathcal{P}]$.

We prove the soundness and completeness of resolution proofs for sequents of the form $\mathcal{P} \vdash \mathcal{G}$.

In Section 5 we provide an elementary semantics for linear logic programs in the setting of Girard quantales [25]. In Appendix B we present the necessary background work in the quantale semantics of linear logic.

In Section 6 we consider the extension of the class of definite formulae to include formulae of the form $!\,D$. This extension introduces very substantial further difficulty in the definition of resolution proof.

We consider the translation of intuitionistic hereditary Harrop formulae in to linear logic, and show that uniform proofs are preserved under this translation.

In Section 7 we consider various issues that arise when considering an implementation of linear logic programming. Foremost amongst these is the question of how to split the antecedent (and succedent) in the $\otimes$-R rule, as discussed earlier. A further question is that of a system of modules. Just as in [20] our system permits modular logic programming via the presence of the implicational formulae $D \multimap G$ in goals, the dependence of a computation being modelled by hypothetical conclusions. The left hand side of the implication must be a definite formula since application of the $\multimap$-R rule, considered as a reduction operator, results in the transfer of the left hand side of the implication to the program. However since the inductive definition of definite formulae permits formulae of the form $G \multimap L$, computations may invoke many modules.

The reader may find it helpful to refer back to this overview to identify the motivation for the various technicalities that follow.

2 Formulae, Definite Formulae and Goal Formulae in Linear Logic

2.1 The Language of Linear Logic

We begin by describing the language of linear logic, [14], [15], [25]. By a *linear predicate language* $\mathcal{L}$ we mean:

- A set of constants $C = \{c, d, \ldots\}$;
- A set of variables $V = \{x, y, \ldots\}$;
- A countable set of function symbols $F = \{f, g, \ldots\}$, equipped with an arity map $\omega_F : F \longrightarrow \mathbb{N}$, where $\mathbb{N}$ denotes the natural numbers;
- A countable set of predicate symbols $P = \{p, q, \ldots\}$, equipped with an arity map $\omega_P : P \longrightarrow \mathbb{N}$;
- The symbols $\mathbf{0}$, $\mathbf{1}$, $\top$, $\bot$, ${}^{\perp}$, $\otimes$, $\oplus$, $\multimap$, $\&$, $⅋$, $\bigwedge$, $\bigvee$.

Henceforth we shall assume that we have some fixed linear predicate language, $\mathcal{L}$. We are now able to define *terms*, *atomic formulae* and *formulae.*

DEFINITION 2.1 (TERMS) Terms *are defined in the usual way:*

- *Constants and variables are terms;*
- *If f is a function symbol of arity $\omega_F(f)$ and if $t_1, \ldots, t_{\omega_F(f)}$ are terms then $ft_1 \ldots t_{\omega_F(f)}$ is a term.*

We denote the universe of terms (of $\mathcal{L}$) by $\mathcal{U}$. □

Note that $\mathcal{U}$ here denotes the set of all terms, including those containing variables, whereas in [20] the same symbol is used to denote the set of all ground terms.

DEFINITION 2.2 (ATOMIC FORMULAE) *We define the* atomic formulae *as follows: if p is a predicate symbol of arity $\omega_P(p)$ and $t_1, \ldots, t_{\omega_P(p)}$ are terms then $pt_1 \ldots t_{\omega_P(p)}$ is an atomic formula. We denote the set of atomic linear formulae of $\mathcal{L}$ by $\mathcal{A}(\mathcal{L})$.* □

DEFINITION 2.3 (FORMULAE) *We define the* formulae *of linear logic as follows:*

- $\mathbf{0}$, $\mathbf{1}$, $\top$ *and* $\perp$ *are formulae;*
- *Atomic formulae are formulae;*
- *If ϕ and ψ are formulae then $\phi \otimes \psi$ is a formula;*
- *If ϕ and ψ are formulae then $\phi \oplus \psi$ is a formula;*
- *If ϕ and ψ are formulae then $\phi \mathbin{⅋} \psi$ is a formula;*
- *If ϕ and ψ are formulae then $\phi \mathbin{\&} \psi$ is a formula;*
- *If ϕ and ψ are formulae then $\phi \multimap \psi$ is a formula;*
- *If ϕ is a formula then $\bigwedge x \,.\, \phi$ is a formula;*
- *If ϕ is a formula then $\bigvee x \,.\, \phi$ is a formula;*
- *If ϕ is a formula then $\phi^\perp$ is a formula.*

We denote the set of linear formulae (of $\mathcal{L}$) by $\mathcal{F}(\mathcal{L})$. Also, we write $\phi \multimapboth \psi$ for $\phi \multimap \psi \& \psi \multimap \phi$.
□

The last clause of this definition is the syntactic (linear) negation of a formulae ϕ. Syntactic negation is defined inductively over the structure of formulae.

Definition 2.4 (Negation) Syntactic linear negation *is defined as follows:*

- $\mathbf{0}^{\perp} =_{\mathrm{def}} \top$, $\mathbf{1}^{\perp} =_{\mathrm{def}} \perp$, $\perp^{\perp} =_{\mathrm{def}} \mathbf{1}$, $\top^{\perp} =_{\mathrm{def}} \mathbf{0}$;
- $(\phi \otimes \psi)^{\perp} =_{\mathrm{def}} \phi^{\perp} \maltese \psi^{\perp}$;
- $(\phi \maltese \psi)^{\perp} =_{\mathrm{def}} \phi^{\perp} \otimes \psi^{\perp}$;
- $(\phi \oplus \psi)^{\perp} =_{\mathrm{def}} \phi^{\perp} \,\&\, \psi^{\perp}$;
- $(\phi \,\&\, \psi)^{\perp} =_{\mathrm{def}} \phi^{\perp} \oplus \psi^{\perp}$;
- $(\bigwedge x \,.\, \phi)^{\perp} =_{\mathrm{def}} \bigvee x \,.\, \phi^{\perp}$;
- $(\bigvee x \,.\, \phi)^{\perp} =_{\mathrm{def}} \bigwedge x \,.\, \phi^{\perp}$;
- $(\phi^{\perp})^{\perp} =_{\mathrm{def}} \phi$.

□

We note that if $\mathcal{F} = \{\phi_1, \ldots, \phi_m\}$ is a multiset of linear formulae then we write $\bigotimes_{\phi \in \mathcal{F}} \phi$ to denote the formula $\phi_1 \otimes \ldots \otimes \phi_m$.

The formulae of a linear predicate language can we extended by the *exponentials* "of course" ! and "why not" ?. If ϕ is a formula then so are $!\phi$ and $?\psi$. Syntactic negation can be extended to formulae constructed using ! and ? by $(!\phi)^{\perp} =_{\mathrm{def}} ?\phi^{\perp}$ and $(?\phi)^{\perp} =_{\mathrm{def}} !\phi^{\perp}$.

Linear sequents are expressions of the form $\Gamma \vdash \Delta$ where the *antecedent* Γ and *succedent* Δ can be considered to be multisets of formulae: we stress, however, that an antecedent is characterized proof-theoretically by the tensor product $\otimes$ of its components and that a succedent is characterized proof-theoretically by the tensor sum $\maltese$ of its components. The rules of linear sequent calculus, in two-sided form, are presented in Appendix A.

The cut elimination theorem holds for linear sequent calculus [14].

2.2 Definite Formulae and Goal Formulae

Following Miller *et al.* we begin our study of logic programming and its proof-theoretic foundation by identifying classes of definite formulae and goal formulae.

Our definitions of definite formulae and goal formulae are motivated by considerations that are similar to those of Miller *et al.* [20], [21]. Informally, definite formulae are intended to be the conjunctive, implicational fragment of the language and goal formulae are intended to be the whole language, restricted to be compatible, via the form of implicational clauses, with definite formulae. In linear logic we take the definite fragment to be the conjunctive part of the multiplicative fragment, the conjunctive part of the additive fragment, together with the implicational fragment. More precisely, we make the following definition:

DEFINITION 2.5 (DEFINITE FORMULAE AND GOAL FORMULAE) *Let A range over atomic formulae. We define classes of* positive literals, definite formulae *and* goal formulae *as follows:*

$$
\begin{array}{llcl}
\textit{Positive Literals} & L & ::= & \mathbf{0} \mid \mathbf{1} \mid \bot \mid \top \mid A \\[1ex]
\textit{Definite formulae} & D & ::= & L \mid D \mathbin{\&} D \mid D \otimes D \mid G \multimap L \mid \bigwedge x\,.\,D \\[1ex]
\textit{Goal formulae} & G & ::= & L \mid L^{\perp} \mid G \otimes G \mid G \oplus G \mid G \mathbin{⅋} G \mid G \mathbin{\&} G \\
 & & & \mid D \multimap G \mid \bigwedge x\,.\,G \mid \bigvee x\,.\,G
\end{array}
$$

□

Note that we allow the negation of ***positive literals*** only. This is because negations of arbitrary formulae can destroy the consistency of the syntactic form of definite formulae. For example, if were to allow the negation of arbitrary definite formulae then $(A_1 \otimes A_2)^{\perp}$ would, by definition, be a definite formulae: however, by the definition of syntactic linear negation $(A_1 \otimes A_2)^{\perp}$ is equal to $A_1^{\perp} ⅋ A_2^{\perp}$, which is not a definite formula. We remark that it is possible to extend the class of definite formulae by the clause $!D$ and the class of goal formulae by the clause $?G$. The addition of $!G$ to the class of goal formulae is not possible in the present formulation.[10]

In the corresponding definition for intuitionistic logic the restriction of the implicational clause of definite formulae to be of the form $G \supset A$, where G is a goal formula and A is an *atomic* formula, so that the resolution rule has a suitably deterministic form, is equivalent, proof-theoretically, to the form $G \supset D$, where D is an arbitrary definite formula [21]. In linear logic however the corresponding equivalence — of the form $G \multimap L$ and $G \multimap D$ — does not hold; its proof would require the equivalence of

$$\phi \multimap (\psi \otimes \chi) \quad \text{and} \quad (\phi \multimap \psi) \otimes (\phi \multimap \chi),$$

and these are not equivalent. For a counterexample, let $\phi = \psi = \chi$. Clearly the sequent $\vdash (\phi \multimap \phi) \otimes (\phi \multimap \phi)$ is provable, as an application of $\otimes$-R to the provable sequents $\vdash \phi \multimap \phi$ and $\vdash \phi \multimap \phi$ results in the sequent given. On the other hand, the sequent $\vdash \phi \multimap (\phi \otimes \phi)$ has no proof.

2.3 Programs and Goals

We consider logic programs to be antecedents of linear sequents of the form $D_1, \ldots, D_m$ and goals to succedents of linear sequents of the form $G_1, \ldots, G_n$. It will be convenient to consider programs and goals to be multisets of formulae. Note that we *cannot* use set-theoretic "inferences" to determine logical inferences as may be done in *e.g.*, in intuitionistic logic [24]: in intuitionistic logic an antecedent can be considered to be the

[10]The impossibility of such an extension to the class of goal formulae arises from the failure of a certain permutation property, *q.v.* Section 3.

conjunction of its components and such conjunctions can be considered to be characterized by set-theoretic union. Similarly, succedents can be considered to be the disjunction of their components and such disjunctions can be considered to be characterized by set-theoretic union. The intuitionistic structural rules of contraction and weakening can therefore be considered to be characterized by properties of sets, *cf.* [20]. However, by considering linear antecedents to be multisets of formulae we can eliminate the need to consider explicit uses of the *exchange* rules of linear logic.

DEFINITION 2.6 (PROGRAMS AND GOALS) Programs *and* goals *are given by the grammar:*

$$\begin{array}{lcl} \textit{Programs} \; \mathcal{P} & ::= & D \mid \mathcal{P}, \mathcal{P} \\ \\ \textit{Goals} \quad\; \mathcal{G} & ::= & G \mid \mathcal{G}, \mathcal{G} \end{array}$$

in which we can interpret the program $\mathcal{P} = D_1, \ldots, D_m$ *as the formula* $D_1 \otimes \ldots \otimes D_m$ *and the goal* $\mathcal{G} = G_1, \ldots, G_n$ *as the formula* $G_1 ⅋ \ldots ⅋ G_n$. □

In Section 3 we define an appropriate notion of *uniform proof*: it is an open problem to determine what is the largest class of formulae for which uniform proofs are complete. The reader is referred to recent work of Hodas and Miller [17] and Andreoli and Pareschi [2] for alternative discussions of this point.

A similar problem obtains for intuitionistic logic, *cf.* [16], [20].

3 Uniform Linear Proofs

As we discussed in Section 1, one of the distinguishing features of logic programming is goal-directed search. More precisely, there is a search operation corresponding to each logical connective, and when searching for a proof of a given goal one applies the search operation that corresponds to the outermost connective of that goal, and then to the outermost connective of each subgoal so generated, *etc.*. For Horn clause goals, *i.e.,* existentially quantified conjunctions of atoms, this corresponds to using unification to "delay" the choice of witnesses for the variables, and a strategy for selecting a particular atom as the next subgoal. For a larger language, such as first-order hereditary Harrop formulae [21], we need a richer set of primitive search operations, or *search primitives*[11], so that each distinct connective corresponds to a distinct search primitive.

Clearly it is not difficult to identify the appropriate search operation for a given right rule, and indeed it is clear that such a procedure will be sound. However, in addition, we need to know that this method of searching for proofs is complete: that we are guaranteed not to miss a proof. Hence our problem is not so much to identify the required search primitives for the right rules of linear logic as to show that the goal-directed search methodology is complete. In [21] it was shown how *uniform proofs* are complete for a certain fragment of (intuitionistic) first-order logic. A uniform proof in that context is

[11]These can be considered to be ***reduction operators*** in the sense of Kleene [18].

one in which the outermost connective of the succedent is introduced in the previous step. Thus, in terms of goal-directed search, this means that right rules are applied as early as possible in the proof-search process. In linear logic, not only are the connectives less "symmetric" and hence more problematic, there is an added complication due to the way that succedents may consist of more than one formula. Nevertheless, it is still possible to define a similar notion of uniformity, although it is a little more intricate.

The main novelty is that, under certain circumstances, it may be necessary to apply a left rule in the middle of a sequence of right rules in order to maintain completeness. For example, consider the sequent $\phi \otimes \psi \vdash \phi \otimes \psi$. Clearly there is a proof in which the $\otimes$-R rule precedes the $\otimes$-L rule, *i.e.,*

$$\cfrac{\cfrac{\phi \vdash \phi \qquad\qquad \psi \vdash \psi}{\phi, \psi \vdash \phi \otimes \psi}\ \otimes\text{-R}}{\phi \otimes \psi \vdash \phi \otimes \psi}\ \otimes\text{-L}$$

but we cannot apply the rules in the reverse order, as neither $\vdash \phi$ nor $\phi \otimes \psi \vdash \psi$ is provable. Hence, we cannot always push the $\otimes$-L above the $\otimes$-R rule; but the cases in which this is not possible may be classified.

A similar example occurs for the $C?$-L rule. Consider the sequent $!\phi \vdash \phi \otimes \phi$. Clearly there is a proof in which the $\otimes$-R rule precedes the $C?$-L rule, *i.e.,*

$$\cfrac{\cfrac{\cfrac{\phi \vdash \phi}{!\phi \vdash \phi}\ !\text{-L} \qquad \cfrac{\phi \vdash \phi}{!\phi \vdash \phi}\ !\text{-L}}{!\phi, !\phi \vdash \phi \otimes \phi}\ \otimes\text{-R}}{!\phi \vdash \phi \otimes \phi}\ C?\text{-L}$$

but we cannot apply the rules in the reverse order, as $\vdash \phi$ is not provable. Thus, as in the previous case, we cannot always push the $C?$-L above the $\otimes$-R rule; but the cases in which this is not possible may be classified.

As it happens, these are the only two such cases; in all the others the "outermost connective first" strategy will suffice. We develop formally the notion of a uniform proof in this context, in the manner of [16].

Let ϕ, ψ, χ and ξ range over linear formulae, and let Γ, Γ' *etc.* range over antecedents and Δ, Δ' *etc.* range over succedents. We work with cut-free linear sequent calculus throughout.

PROPOSITION 3.1 *If there is a proof of a given endsequent which has a subproof of the form*

$$\cfrac{\cfrac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'}\; *\text{-R}}{\Gamma'' \vdash \Delta''}\; \natural\text{-L}$$

where $*$-R *is either* $\perp$-R, $\parr$-R, $\oplus$-R, $\multimap$-R, ?-R, *W*?-R, *C*?-R, $\bigwedge$-R *or* $\bigvee$-R, *and* $\natural$-L *is one of either* &-L, $\otimes$-L, !-L, *W*?-L, *C*?-L *or* $\bigwedge$-L, *then there is a proof of the same endsequent with this subproof replaced by a subproof of the form*

$$\cfrac{\cfrac{\Gamma \vdash \Delta}{\Gamma''' \vdash \Delta'''}\; \natural\text{-L}}{\Gamma'' \vdash \Delta''}\; *\text{-R}$$

PROOF
We give the proof only for the cases $\parr$-R and $\otimes$-L, the others are similar. Given a subproof

$$\cfrac{\cfrac{\Gamma, \chi, \xi \vdash \phi, \psi, \Delta}{\Gamma, \chi, \xi \vdash \phi \parr \psi, \Delta}\; \parr\text{-R}}{\Gamma, \chi \otimes \xi \vdash \phi \parr \psi, \Delta}\; \otimes\text{-L}$$

it is clear that there is a subproof

$$\cfrac{\cfrac{\Gamma, \chi, \xi \vdash \phi, \psi, \Delta}{\Gamma, \chi \otimes \xi \vdash \phi, \psi, \Delta}\; \otimes\text{-L}}{\Gamma, \chi \otimes \xi \vdash \phi \parr \psi, \Delta}\; \parr\text{-R}$$

□

Note that this proposition does not hold for the rule $\perp$-L. For example, consider the sequent $\phi, (\phi \oplus \psi)^{\perp} \vdash$, which is provable in the linear sequent calculus, as demonstrated by the proof below.

$$\cfrac{\cfrac{\phi \vdash \phi}{\phi \vdash \phi \oplus \psi}\; \oplus\text{-R}}{\phi, (\phi \oplus \psi)^{\perp} \vdash}\; \perp\text{-L}$$

However we cannot apply the rules in the reverse order to derive this sequent. This is one of the reasons that we do not allow formulae such as $\phi^{\perp}$ to appear in programs.[12].

[12]Unless, of course, it appears in a goal.

We shall see later another important permutation property fails for this rule.

PROPOSITION 3.2 *If there is a proof of a given endsequent in which there is a subproof of the form*

$$\dfrac{\Gamma'' \vdash \phi, \Delta'' \qquad \dfrac{\Gamma, \psi \vdash \Delta}{\Gamma', \psi, \vdash \Delta'}\, *\text{-R}}{\Gamma'', \Gamma', \phi \multimap \psi \vdash \Delta'', \Delta'}\, \multimap\text{-L}$$

where $*$-R *is one of* $\bot$-R, $\parr$-R, $\otimes$-R, $\multimap$-R, ?-R, *W*?-R, *C*?-R, $\bigwedge$-R, $\bigvee$-R, *then there is a proof of the same endsequent with this subproof replaced by a subproof of the form*

$$\dfrac{\dfrac{\Gamma'' \vdash \phi, \Delta'' \qquad \Gamma, \psi, \vdash \Delta}{\Gamma'', \Gamma, \phi \multimap \psi \vdash \Delta'', \Delta}\, \multimap\text{-L}}{\Gamma'', \Gamma', \phi \multimap \psi \vdash \Delta'', \Delta'}\, *\text{-R}$$

PROOF
We give the proof only for the case $\parr$-R. The others are similar. Given a subproof

$$\dfrac{\Gamma'' \vdash \phi, \Delta'' \qquad \dfrac{\Gamma, \psi \vdash \chi, \xi, \Delta}{\Gamma', \psi, \vdash \chi \parr \xi, \Delta'}\, \parr\text{-R}}{\Gamma'', \Gamma', \phi \multimap \psi \vdash \chi \parr \xi, \Delta'', \Delta'}\, \multimap\text{-L}$$

it is clear that there is a subproof

$$\dfrac{\dfrac{\Gamma'' \vdash \phi, \Delta'' \qquad \Gamma, \psi, \vdash \chi, \xi, \Delta}{\Gamma'', \Gamma', \phi \multimap \psi \vdash \chi, \xi, \Delta'', \Delta'}\, \multimap\text{-L}}{\Gamma'', \Gamma', \phi \multimap \psi \vdash \chi \parr \xi, \Delta'', \Delta'}\, \parr\text{-R}$$

□

PROPOSITION 3.3 *If there is a proof of a given endsequent in which there is a subproof of the form*

$$\dfrac{\dfrac{\Gamma \vdash \phi, \Delta \qquad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \,\&\, \psi, \Delta}\, \&\text{-R}}{\Gamma' \vdash \phi \,\&\, \psi, \Delta}\, \natural\text{-L}$$

where $\natural$-L *is either* &-L, $\otimes$-L, !-L, *W*?-L, *C*?-L *or* $\bigwedge$-L, *then there is a proof of the same endsequent with this subproof replaced by a subproof of the form*

$$\cfrac{\cfrac{\Gamma \vdash \phi, \Delta}{\Gamma' \vdash \phi, \Delta}\ \sharp\text{-L} \qquad \cfrac{\Gamma \vdash \psi, \Delta}{\Gamma' \vdash \psi, \Delta}\ \sharp\text{-L}}{\Gamma' \vdash \phi \,\&\, \psi, \Delta}\ \&\text{-R}$$

PROOF
We give the proof only for the case $\otimes$-L. The others are similar. Given a subproof

$$\cfrac{\cfrac{\Gamma, \chi, \xi \vdash \phi, \Delta \qquad \Gamma, \chi, \xi \vdash \psi, \Delta}{\Gamma, \chi, \xi \vdash \phi \,\&\, \psi, \Delta}\ \&\text{-R}}{\Gamma, \chi \otimes \xi \vdash \phi \,\&\, \psi, \Delta}\ \otimes\text{-L}$$

it is clear that there is a subproof

$$\cfrac{\cfrac{\Gamma, \chi, \xi \vdash \phi, \Delta}{\Gamma, \chi \otimes \xi \vdash \phi, \Delta}\ \otimes\text{-L} \qquad \cfrac{\Gamma, \chi, \xi \vdash \psi, \Delta}{\Gamma, \chi \otimes \xi \vdash \psi, \Delta}\ \otimes\text{-L}}{\Gamma, \chi \otimes \xi \vdash \phi \,\&\, \psi, \Delta}\ \&\text{-R}$$

□

PROPOSITION 3.4 *If there is proof of a given endsequent in which there is a subproof of the form*

$$\cfrac{\Gamma' \vdash \chi, \Delta' \qquad \cfrac{\Gamma, \xi \vdash \phi, \Delta \qquad \Gamma, \xi \vdash \phi, \Delta}{\Gamma, \xi \vdash \phi \,\&\, \psi, \Delta}\ \&\text{-R}}{\Gamma, \Gamma', \chi \multimap \xi \vdash \phi \,\&\, \psi, \Delta', \Delta}\ \multimap\text{-L}$$

then there is a proof of the same endsequent with this subproof replaced by a subproof of the form

$$\cfrac{\cfrac{\Gamma' \vdash \chi, \Delta \qquad \Gamma, \xi \vdash \phi, \Delta}{\Gamma', \Gamma, \chi \multimap \xi \vdash \phi, \Delta, \Delta'}\ \multimap\text{-L} \qquad \cfrac{\Gamma' \vdash \chi, \Delta' \qquad \Gamma, \xi \vdash \psi, \Delta}{\Gamma', \Gamma, \chi \multimap \xi \vdash \psi, \Delta, \Delta'}\ \multimap\text{-L}}{\Gamma', \Gamma, \chi \multimap \xi, \Gamma \vdash A \,\&\, \psi, \Delta, \Delta'}\ \&\text{-R}$$

PROOF
Obvious. □

COROLLARY 3.5 *If there is a proof of a given endsequent in which there is a subproof in which either* $\perp$-R, $\maltese$-R, $\oplus$-R, $\multimap$-R, ?-R, *W*?-R, *C*?-R, $\bigwedge$-R, $\bigvee$-R *or* &-R *precedes* *either* &-L, ⊗-L, !-L, *W*?-L, *C*?-L, ⋀-L *or precedes* ⊸-L *on the right, then there is proof of the same endsequent in which this subproof is replaced by a subproof in which the left rule precedes the right rule.*

PROOF
Follows immediately from Propositions 3.1, 3.2, 3.3 and 3.4. □

PROPOSITION 3.6 *If there is a proof of a given endsequent in which there is a subproof of the form*

$$\cfrac{\cfrac{\Gamma \vdash \phi, \Delta \qquad \Gamma' \vdash \psi, \Delta'}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}}{\Gamma'', \Gamma''' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \sharp\text{-L}$$

where ♯-L *is either* &-L, !-L, *W*?-L *or* ⋀-L, *then there is a proof of the same ensequent in which this subproof is replaced by a subproof of the form*

$$\cfrac{\cfrac{\Gamma \vdash \phi, \Delta}{\Gamma'' \vdash \phi, \Delta}\ \sharp\text{-L} \qquad \cfrac{\Gamma' \vdash \psi, \Delta'}{\Gamma''' \vdash \psi, \Delta'}\ \sharp\text{-L}}{\Gamma'', \Gamma''' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}$$

PROOF
We give the proof only for the case &-L. The others are similar.

Without loss of generality, we assume that the &-L rule is applied to the sequent $\Gamma \vdash \phi, \Delta$ rather than $\Gamma' \vdash \psi, \Delta'$. Given a subproof

$$\cfrac{\cfrac{\Gamma, \chi \vdash \phi, \Delta \qquad \Gamma' \vdash \psi, \Delta'}{\Gamma, \chi, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}}{\Gamma, \chi \,\&\, \xi \vdash \phi \otimes \psi, \Delta, \Delta'}\ \&\text{-L}$$

it is clear that there is a subproof

$$\cfrac{\cfrac{\Gamma, \chi, \vdash \phi, \Delta}{\Gamma, \chi \,\&\, \xi \vdash \phi, \Delta}\ \&\text{-L} \qquad \Gamma' \vdash \psi, \Delta'}{\Gamma, \chi \,\&\, \xi \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}$$

□

PROPOSITION 3.7 *If there is a proof of a given endsequent in which there is a subproof of the form*

$$\dfrac{\Gamma''\vdash\chi,\Delta'' \qquad \dfrac{\Gamma,\xi\vdash\Delta \qquad \Gamma'\vdash\Delta'}{\Gamma,\xi,\Gamma'\vdash\Delta'''}\,\otimes\text{-R}}{\Gamma'',\Gamma',\Gamma,\chi\multimap\xi\vdash\Delta'',\Delta'''}\,\multimap\text{-L}$$

then there is a proof of the same endsequent with this subproof replaced by a subproof of the form

$$\dfrac{\dfrac{\Gamma''\vdash\chi,\Delta'' \qquad \Gamma,\xi\vdash\Delta}{\Gamma'',\Gamma,\chi\multimap\xi\vdash\Delta'',\Delta}\,\multimap\text{-L} \qquad \Gamma'\vdash\Delta'}{\Gamma'',\Gamma',\Gamma,\chi\multimap\xi\vdash\Delta'',\Delta'''}\,\otimes\text{-R}$$

PROOF
Obvious. □

COROLLARY 3.8 *If there is a proof of a given endsequent in which there is a subproof in which* ⊗-R *precedes either* &-L, !-L, *W*?-L, $\bigwedge$-L *or precedes* ⊸ -L *on the right, then there is a proof of the same endsequent with this subproof replaced by a subproof in which the left rule precedes the right rule.*

PROOF
Follows immediately from Propositions 3.6 and 3.7.

PROPOSITION 3.9 *If there is a proof of a given endsequent in which there is a subproof of the form*

$$\dfrac{\dfrac{\Gamma\vdash\phi,\Delta \qquad \Gamma'\vdash\psi,\Delta'}{\Gamma,\Gamma'\vdash\phi\otimes\psi,\Delta,\Delta'}\,\otimes\text{-R}}{\Gamma'',\chi\otimes\xi\vdash\phi\otimes\psi,\Delta,\Delta'}\,\otimes\text{-L}$$

where $\chi,\xi\in\Gamma\cup\Gamma'$, *then if* $\chi,\xi\in\Gamma$ *or* $\chi,\xi\in\Gamma'$ *there is a proof of the same endsequent with this subproof replaced by a subproof in which the* ⊗-L *rule precedes the* ⊗-R *rule.*

PROOF
Without loss of generality we may assume that $\chi,\xi\in\Gamma$. It should be clear that we then have a subproof of the following form

$$\dfrac{\dfrac{\Gamma\vdash\phi,\Delta}{\Gamma''',\chi\otimes\xi\vdash\phi,\Delta}\,\otimes\text{-L} \qquad \Gamma'\vdash\psi,\Delta'}{\Gamma'',\chi\otimes\xi\vdash\phi\otimes\psi,\Delta,\Delta'}\,\otimes\text{-R}$$

□

Note that the restriction in the previous proposition is necessary in that there are some cases in which the $\otimes$-R rule must precede the $\otimes$-L rule in order for a proof to exist. For example, consider the sequent $\phi \otimes \psi \vdash \phi \otimes \psi$. Clearly there is a proof in which the $\otimes$-R rule precedes the $\otimes$-L rule, *i.e.,*

$$\dfrac{\dfrac{\phi \vdash \phi \qquad\qquad \psi \vdash \psi}{\phi, \psi \vdash \phi \otimes \psi}\ \otimes\text{-R}}{\phi \otimes \psi \vdash \phi \otimes \psi}\ \otimes\text{-L}$$

but we cannot apply the rules in the reverse order, as neither $\vdash \phi$ nor $\phi \otimes \psi \vdash \psi$ is provable. Hence, we cannot always push the $\otimes$-L above the $\otimes$-R rule; but the cases in which this is not possible may be classified.

PROPOSITION 3.10 *If there is a proof of a given endsequent in which there is a subproof of the form*

$$\dfrac{\dfrac{\Gamma \vdash \phi, \Delta \qquad\qquad \Gamma' \vdash \psi, \Delta'}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}}{\Gamma'', !\chi \vdash \phi \otimes \psi, \Delta, \Delta'}\ C?\text{-L}$$

where $\{!\chi, !\chi\} \in \Gamma \cup \Gamma'$, *then if* $\{!\chi, !\chi\} \subseteq \Gamma$ *or* $\{!\chi, !\chi\} \subseteq \Gamma'$ *there is a proof of the same endsequent with this subproof replaced by a subproof in which the* $\otimes$-L *rule precedes the* $\otimes$-R *rule.*

PROOF
Without loss of generality we may assume that $\{!\chi, !\chi\} \subseteq \Gamma$. It should be clear that we then have a subproof of the following form

$$\dfrac{\dfrac{\Gamma \vdash \phi, \Delta}{\Gamma''', !\chi \vdash \phi, \Delta}\ C?\text{-L} \qquad \Gamma' \vdash \psi, \Delta'}{\Gamma'', !\chi \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}$$

□

As in the previous case, this restriction is necessary. For example, consider the sequent $!\phi \vdash \phi \otimes \phi$. Clearly there is a proof in which the $\otimes$-R rule precedes the $C?$-L rule, *i.e.,*

$$\dfrac{\dfrac{\dfrac{\phi \vdash \phi}{!\phi \vdash \phi}\ !\text{-L} \qquad \dfrac{\phi \vdash \phi}{!\phi \vdash \phi}\ !\text{-L}}{!\phi, !\phi \vdash \phi \otimes \phi}\ \otimes\text{-R}}{!\phi \vdash \phi \otimes \phi}\ C?\text{-L}$$

However, we cannot apply the rules in the reverse order, as $\vdash \phi$ is not provable. Thus, as in the previous case, we cannot always push the $C?$-L above the $\otimes$-R rule; but the cases in which this is not possible may be classified.

Definition 3.11 (RL, LR and Locally LR Proofs) *We define the following shapes of linear proofs:*

1. *A proof is* RL *if there is an occurrence of a right rule preceding a left rule;*
2. *A proof is* LR *if all occurrences of right rules appear after all left rules;*
3. *A proof is* locally LR *if the only occurrences of a right rule preceding a left rule are either of the form*

$$\cfrac{\cfrac{\Xi_1}{\Gamma \vdash \phi, \Delta}\ *\text{-R} \qquad \cfrac{\Xi_2}{\Gamma', \psi \vdash \Delta'}\ \sharp\text{-L}}{\Gamma, \Gamma', \phi \multimap \psi \vdash \Delta, \Delta'}\ \multimap\text{-L}$$

where Ξ_1 *and* Ξ_2 *are locally* LR, *or of the form*

$$\cfrac{\cfrac{\cfrac{\Xi_1}{\Gamma \vdash \phi, \Delta} \qquad \cfrac{\Xi_2}{\Gamma' \vdash \psi, \Delta'}}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}}{\Gamma'', \chi \otimes \xi \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-L}$$

where $\chi \in \Gamma$, $\xi \in \Gamma'$, *and* Ξ_1 *and* Ξ_2 *are locally* LR, *or of the form*

$$\cfrac{\cfrac{\cfrac{\Xi_1}{\Gamma \vdash \phi, \Delta} \qquad \cfrac{\Xi_2}{\Gamma' \vdash \psi, \Delta'}}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}}{\Gamma'', !\chi \vdash \phi \otimes \psi, \Delta, \Delta'}\ C?\text{-L}$$

where $!\chi \in \Gamma$, $!\chi \in \Gamma'$, *and* Ξ_1 *and* Ξ_2 *are locally* LR. □

The next proposition, and its corollary, give the completeness of locally LR proofs with respect to a certain class of linear sequent calculus proofs, namely those in which there are no occurrences of the rules $\bot$-L, $\oplus$-L, ⅋-L, ?-L and $\bigvee$-L. At this point, and henceforth, it is crucial that we assume that *negation is restricted to atomic formulae (and constants) only.* If we do not make this assumption then we can can force the contradiction of the condition that proof contain no instances of the $\oplus$-L, $\parr$-L, ?-L and $\bigvee$-L rules. For example, the rule:

$$\frac{\Gamma, p^{\perp} \parr q^{\perp} \vdash \Delta}{\Gamma \vdash p \otimes q, \Delta},$$

is derived from an instance of $\perp$-R by the definition of syntactic linear negation.

PROPOSITION 3.12 *If* $\Gamma \vdash \Delta$ *has an* RL *proof not containing* !-R, $\perp$-L, $\oplus$-L, $\parr$-L, ?-L *or* $\bigvee$-L, *then* $\Gamma \vdash \Delta$ *has a locally* LR *proof.*

PROOF
Consider the occurrence of a right rule preceding a left rule which is closest to the root of the proof tree. If the right rule occurs on the left of $\multimap$-L, we are done. Otherwise, if the right rule is not $\otimes$-R, then by Corollary 3.5 there is a proof with the left rule preceding the right one.

Hence let the right rule be $\otimes$-R. If the left rule is neither $\otimes$-L nor C?-L, by Corollary 3.8 there is a proof with the left rule preceding the right one.

If the left rule is $\otimes$-L, the proof is of the form

$$\cfrac{\cfrac{\cfrac{\Xi_1}{\Gamma \vdash \phi, \Delta} \qquad \cfrac{\Xi_2}{\Gamma' \vdash \psi, \Delta'}}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}}{\Gamma'', \chi \otimes \xi \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-L}$$

where $\chi, \xi \in \Gamma \cup \Gamma'$.

Now if $\chi \in \Gamma$ and $\xi \in \Gamma'$ we are done. Clearly a similar result holds when $\chi \in \Gamma'$ and $\xi \in \Gamma$.

Hence, without loss of generality, we assume $\chi, \xi \in \Gamma$, and so $\Gamma = \{\chi\} \cup \{\xi\} \cup \Gamma'''$ and $\Gamma'' = \Gamma' \cup \Gamma'''$. Then there is a proof of the form

$$\cfrac{\cfrac{\cfrac{\Xi_1}{\Gamma \vdash \phi, \Delta}}{\Gamma''', \chi \otimes \xi \vdash \phi, \Delta}\ \otimes\text{-L} \qquad \cfrac{\Xi_2}{\Gamma' \vdash \psi, \Delta'}}{\Gamma'', \chi \otimes \xi \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}$$

Hence let the left rule be C?-L, and so the proof is of the form

$$\dfrac{\dfrac{\dfrac{\Xi_1}{\Gamma \vdash \phi, \Delta} \qquad \dfrac{\Xi_2}{\Gamma' \vdash \psi, \Delta'}}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}}{\Gamma'', !\chi \vdash \phi \otimes \psi, \Delta, \Delta'}\ C?\text{-L}$$

where $\{!\chi, !\chi\} \subseteq \Gamma \cup \Gamma'$.

Now if $!\chi \in \Gamma$ and $!\chi \in \Gamma'$ we are done. Clearly a similar result holds when $!\chi \in \Gamma'$ and $!\chi \in \Gamma$.

Hence, without loss of generality, we assume $\{!\chi, !\chi\} \subseteq \Gamma$, and so $\Gamma = \{!\chi\} \cup \{!\chi\} \cup \Gamma'''$ and $\Gamma'' = \Gamma' \cup \Gamma'''$. Then there is a proof of the form

$$\dfrac{\dfrac{\dfrac{\Xi_1}{\Gamma \vdash \phi, \Delta}}{\Gamma''', !\chi \vdash \phi, \Delta}\ C?\text{-L} \qquad \dfrac{\Xi_2}{\Gamma' \vdash \psi, \Delta'}}{\Gamma'', !\chi \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}$$

We may then repeat this process for the next rule application (towards the root of the proof tree), and so on until the resulting subproof is locally LR. We may perform this operation for all occurrences of a right rule which immediately precedes a left rule, and hence obtain a proof which is locally LR. □

COROLLARY 3.13 *If there is a proof of* $\Gamma \vdash \Delta$ *which contains no occurrences of* !-R, $\bot$-L, $\oplus$-L, ⅋ -L, ?-L *or* $\bigvee$-L, *then there is a locally* LR *proof of* $\Gamma \vdash \Delta$.

PROOF
If the proof of $\Gamma \vdash \Delta$ is LR, then it is locally LR and we are done. Otherwise, the proof is RL, and by the above proposition, $\Gamma \vdash \Delta$ has a locally LR proof. □

Thus we find that locally LR proofs are complete for sequents of the form $\mathcal{P} \vdash \mathcal{G}$, where $\mathcal{P}$ is a program and $\mathcal{G}$ is a goal. We will sometimes refer to locally LR proofs as *uniform* proofs, in order to emphasize the similarities between this class of proofs and the uniform proofs of Miller *et al.* [21]. Corollary 3.13 shows that we may think of uniform proofs as a normal form proofs in our fragment of linear logic.

However, locally LR proofs are not quite satisfactory for our purposes, as they might involve some inessentially complicated subproofs on the right hand side of the $\multimap$-L rule. Hence we introduce the notion of a *simple* proof, *cf.* [20].

The intuition behind simple proofs is that the right hand subproof used in the $\multimap$-L rule should be trivial, *i.e.*, require no effort to prove. For example the following proof is not simple:

$$\dfrac{\chi \vdash \chi \qquad \dfrac{\phi \vdash \phi}{\phi \,\&\, \psi \vdash \phi}\ \&\text{-L}}{\chi, \chi \multimap \phi \,\&\, \psi \vdash \phi}\ \multimap\text{-L}$$

Note also that a sequent $\mathcal{P} \vdash \mathcal{G}$, where $\mathcal{P}$ is a multiset of definite formula and $\mathcal{G}$ is a multiset of goal formulae, might also have a proof which is not simple. The following proof is such an example.

$$\dfrac{\phi \vdash \phi \qquad \dfrac{\phi \vdash \phi \qquad \psi \vdash \psi}{\phi, \phi \multimap \psi \vdash \psi}\ \multimap\text{-L}}{\phi, \phi \multimap \phi, \phi \multimap \psi \vdash \psi}\ \multimap\text{-L}$$

However, it is clear that the sequent $\phi, \phi \multimap \phi, \phi \multimap \psi \vdash \psi$ does have a simple proof. This property in fact holds for any sequent in which the antecedent is a program and the consequent a goal. First we present a necessary technical result.

DEFINITION 3.14 (SIMPLE PROOFS) *A linear proof Ξ is said to be simple if every for occurrence in Ξ of the $\multimap$-L rule*

$$\begin{array}{c} \Xi_1 \qquad\qquad \Xi_2 \\ \dfrac{\Gamma \vdash \phi, \Delta \qquad \Gamma', \psi \vdash \Delta'}{\Gamma, \phi \multimap \psi, \Gamma' \vdash \psi, \Delta, \Delta'} \\ \Xi_3 \end{array}$$

the right upper subproof Ξ_2, *with root* $\Gamma', \psi \vdash \Delta'$, *is of height* 1; *that is it consists of just one (axiom) inference.* □

PROPOSITION 3.15 *Let Ξ be a proof of $\Gamma \vdash \Delta$. If there is a subproof in Ξ of the form*

$$\dfrac{\Gamma'' \vdash \psi, \Delta'' \qquad \dfrac{\Gamma, \phi \vdash \Delta}{\Gamma', \phi, \vdash \Delta'}\ *\text{-L}}{\Gamma'', \Gamma', \psi \multimap \phi \vdash \Delta'', \Delta'}\ \multimap\text{-L}$$

where $*$-L *is one of* $\&$ -L, $\otimes$-L, !-L, $W?$-L, $C?$-L, $\bigwedge$-L, *then there is a proof of the form*

$$\dfrac{\dfrac{\Gamma'' \vdash \psi, \Delta'' \qquad \Gamma, \phi, \vdash \Delta}{\Gamma'', \Gamma, \psi \multimap \phi \vdash \Delta'', \Delta}\ \multimap\text{-L}}{\Gamma'', \Gamma', \psi \multimap \phi \vdash \Delta'', \Delta'}\ *\text{-L}$$

PROOF
We only give the proof for the case $\otimes$-L. The others are similar. Given a subproof

$$\dfrac{\Gamma' \vdash \psi, \Delta' \qquad \dfrac{\Gamma, \phi, \chi, \xi \vdash \Delta}{\Gamma, \phi, \chi \otimes \xi \vdash \Delta}\ \otimes\text{-L}}{\Gamma', \Gamma, \psi \multimap \phi, \chi \otimes \xi \vdash \Delta', \Delta}\ \multimap\text{-L}$$

it is clear that there is a subproof

$$\dfrac{\dfrac{\Gamma' \vdash \psi, \Delta' \qquad \Gamma, \chi, \xi, \phi \vdash \Delta}{\Gamma', \Gamma, \chi, \xi, \psi \multimap \phi \vdash \Delta', \Delta}\ \multimap\text{-L}}{\Gamma', \Gamma, \chi \otimes \xi, \psi \multimap \phi \vdash \Delta', \Delta}\ \otimes\text{-L}$$

This completes the proof. □

Note that this proposition fails for the $\perp$-L rule. For example, consider the proof

$$\dfrac{q \vdash q \qquad \dfrac{\dfrac{q \vdash q \qquad p \vdash p}{q,\ q \multimap p \vdash p}\ \multimap\text{-L}}{q,\ q \multimap p,\ p^{\perp} \vdash}\ \perp\text{-L}}{q,\ q,\ q \multimap p,\ q \multimap p^{\perp} \vdash}\ \multimap\text{-L}$$

There is no way to push the occurrence of $\multimap$-L any earlier in the proof, as then the required formula cannot be constructed. This is the most important reason why we limit the appearance of linear negation to goals, so that we need only use simple proofs. This restriction also means that the process of searching for a proof will be goal-directed; for example, the sequent $p, p^{\perp} \vdash$ is provable, but clearly as the goal is empty, the notion of goal-dericted search cannot be applied.

Next we show that any LR proof can be converted into a simple LR proof.

PROPOSITION 3.16 *Let Ξ be a proof of $\Gamma \vdash \Delta$ in which there are no occurrences of* !-R, $\perp$-L, ⅋-L, $\oplus$-L, $\bigvee$-L *or* ?-L. *If Ξ is locally LR, then $\Gamma \vdash \Delta$ has a simple locally LR proof.*

PROOF
We proceed by induction on the number of occurrences of the $\multimap$-L rule in the proof. The base case corresponds to the occurrence(s) of $\multimap$-L which appear closest to the leaves of the proof tree.

Consider the right hand sequent in the occurrence of $\multimap$-L. If this is initial, then we are done. Otherwise, consider the rule immediately preceding it. If this is a right rule, then by Propositions 3.2, 3.4 and 3.7 we know that we may replace this subproof with an equivalent one in which the $\multimap$-L rule precedes the right rule. Otherwise the preceding rule must be either ⊗-L, & -L, !-L or $\bigwedge$-L, and by Proposition 3.15 we know that we may similarly replace this subproof with an equivalent one in which the $\multimap$-L rule precedes the (other) left rule. Clearly we may repeat this process for each rule preceding the $\multimap$-L rule until there is no preceding rule, ***i.e.,*** that the occurrence is simple. Note that the sequence of the other rules is not changed, so that the resulting proof is still locally LR.

Hence we assume that given an occurrence of the $\multimap$-L rule, all previous occurrences (*i.e.,* those towards the leaves of the tree) are simple. Now as above, if the occurrence of the $\multimap$-L rule is simple, we are done. Otherwise consider the previous rule on the right of the $\multimap$-L rule. As above, if it is a right rule or any of ⊗-L, & -L, !-L or $\bigwedge$-L, then there is a subproof with the order of the rules reversed. We may then proceed as above until we either arrive at an initial sequent or we find that the rule preceding $\multimap$-L on the rule is another occurrence of $\multimap$-L. By the hypothesis, we know that this must be a simple occurrence, and the proof is of the form

$$\dfrac{\Gamma' \vdash \psi, \Delta' \qquad \dfrac{\Gamma, \chi \vdash \phi, \Delta \qquad L \vdash L}{\Gamma, \chi, \phi \multimap L \vdash \Delta, L}\ \multimap\text{-L}}{\Gamma', \Gamma, \phi \multimap L, \psi \multimap \chi \vdash \Delta, \Delta', L}\ \multimap\text{-L}$$

which clearly we may replace with a proof of the form

$$\dfrac{\dfrac{\Gamma' \vdash \psi, \Delta' \quad \Gamma, \chi \vdash \phi, \Delta}{\Gamma', \Gamma, \psi \multimap \chi \vdash \psi, \Delta, \Delta'}\ \multimap\text{-L} \qquad L \vdash L}{\Gamma', \Gamma, \phi \multimap L, \psi \multimap \chi \vdash \Delta, \Delta', L}\ \multimap\text{-L}$$

in which the latter application is simple, and by the hypothesis the former application may also be made simple. Note that this reordering of the proof also preserves the property that the proof is locally LR.

Hence, any locally LR proof can be made into a simple locally LR proof. □

LEMMA 3.17 *Let $\mathcal{P}$ be a program and $\mathcal{G}$ be a goal. A proof of $\mathcal{P} \vdash \mathcal{G}$ contains no occurrences of* !-R, ⊥-L, ⅋-L, ⊕-L, ?-L *or* $\bigvee$-L.

PROOF
A simple induction on the structure of proofs. □

Thus we arrive at the following theorem:

THEOREM 3.18 *Let Γ be a multiset of definite formulae, and let Δ be a multiset of goal formulae. Then $\Gamma \vdash \Delta$ has a linear proof if and only if $\Gamma \vdash \Delta$ has a simple locally LR proof.*

PROOF
By Lemma 3.17 we have that any linear proof of $\Gamma \vdash \Delta$ doesn't contain ⊥-L, ⅋-L, ⊕-L,

?-L, !-L or $\bigvee$-L; and so by Corollary 3.13, $\Gamma \vdash \Delta$ has a locally LR proof. Hence by Proposition 3.16 $\Gamma \vdash \Delta$ has a simple locally LR proof. □

Note that Theorem 3.18 also holds when Γ contains formulae of the form $!D$, where D is a definite formula, *q.v.* Proposition 6.3.

Thus we need only consider locally LR proofs which are simple.

4 Resolution

We saw in the previous section how locally LR proofs are complete for sequents $\mathcal{P} \vdash \mathcal{G}$ where $\mathcal{P}$ is a multiset of definite formulae and $\mathcal{G}$ is a multiset of goal formulae. In this section we show how a more specific class of proofs, which we call *resolution* proofs, may be used in a similar fashion. The important characteristics of resolution proofs are that they are *goal directed* and use just one left rule, namely *resolution.* First we introduce the notions of *clause* and *clausal decomposition* of a program P.

Definition 4.1 (Clause) *A* linear clause *is a formula of the form L or $G \multimap L$, where L ranges over positive literals and G ranges over goal formulae.* □

Note that when G is a conjunction of atoms, a linear clause in the above sense is just a *Horn* clause.

Below we show how to generate a multiset of clauses from a multiset of definite formulae via the mapping [-].

Definition 4.2 (Clausal Decomposition) *Let $\cup$ denote multiset union. We define a mapping $[-]$ from definite formulae to multisets of definite formulae inductively as follows:*

$$\begin{array}{lll}
[\mathcal{P}] & =_{\text{def}} & \bigcup_{D\in\mathcal{P}}[D] \\
[L] & =_{\text{def}} & \{L\} \\
[D_1 \,\&\, D_2] & =_{\text{def}} & [D_1] \text{ or } [D_2] \\
[D_1 \otimes D_2] & =_{\text{def}} & [D_1] \cup [D_2] \\
[G \multimap L] & =_{\text{def}} & \{G \multimap L\} \\
[\bigwedge x \,.\, D] & =_{\text{def}} & [D].
\end{array}$$

□

Note that $[\mathcal{P}]$ is a multiset of linear clauses.[13] For example, let $\mathcal{P}$ be given by the single definite formula

$$\bigwedge x \,.\, (p(x) \otimes (p(x) \multimap q(x)))$$

[13]The decomposition [-] is related to the theory of *proof nets*, and indeed to the notions of *reduction ordering* — a combination of *subformula ordering* and *substitution ordering* — and *matrix methods,* described in [3], [7], [24] and [23]. Such constructions are useful in the study of Permutation Theorems, *q.v.* Theorem 3.3.

then we have that $[\mathcal{P}]$ is the multiset

$$\{p(x), p(x) \multimap q(x)\}.$$

Thus we may think of the transformation $[-]$ as a mapping which given a program, *i.e.,* a multiset of definite formulae, yields a multiset of clauses which are the *compiled* form of the program, *i.e.,* the form for which resolution proof can be defined. For given $\mathcal{P}$, $[\mathcal{P}]$ can be considered to be a *normal form* of $\mathcal{P}$.

Note that there are free variables in the clauses in $[\mathcal{P}]$; these are the variables that are "outermost" in $\mathcal{P}$. Now we may think of the scope of these free variables as being "global", in that all occurrences of the variable must be updated consistently. Due to the possibility of splitting the program during computation, this may mean updating variables simultaneously across several branches of the proof.[14] This leads us to the definitions that follow.

DEFINITION 4.3 (DISTINCTNESS) *Let $\mathcal{D}$ be a multiset of definite formulae. We say $\mathcal{D}'$ and $\mathcal{D}''$ are* distinct copies of $\mathcal{D}$ *if $\mathcal{D}'$ and $\mathcal{D}''$ are obtained from $\mathcal{D}$ by renaming the free variables of $\mathcal{D}$ so that $\mathcal{D}'$ and $\mathcal{D}''$ have no free variable names in common.* □

We come now to the definition of resolution proof.

DEFINITION 4.4 (RESOLUTION PROOF) *We define the notion of* resolution proof *by defining a relation $\vdash_R$. Let $\mathcal{D}$ range over multisets of definite formulae and let $\cup$ denote multiset union. A* resolution proof *is tree regulated by the following rules, which is constructed from root to leaves, and is such that the resolution rule is applied only when no other rule is applicable:*

1. *The axiom judgement is given by:*

$$\frac{}{\{L'\} \vdash_R L}$$

where $L = L'[\vec{t}/\vec{x}]$;

2. *We shall refer to this rule as the* resolution *rule:*

$$\frac{\mathcal{D}[\vec{t}/\vec{x}] \vdash_R G, \mathcal{G} \qquad \{L\} \vdash_R L}{\mathcal{D} \cup \{G' \multimap L'\} \vdash_R L, \mathcal{G}}$$

where $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$;

[14]In terms of the resolution proofs defined below, those branches that are above the same premise of the last & –rule (Rule 7 in the definition of resolution proof, below) as the sequent to which the resolution rule is applied. The form of the &-rule in such proofs enforces the restriction of substitutions to their own &-branches.

3. The $\perp$-rule:

$$\frac{\mathcal{D} \cup \{L\} \vdash_R \mathcal{G}}{\mathcal{D} \vdash_R L^{\perp}, \mathcal{G}}$$

4. The $\otimes$-rule:

$$\frac{\mathcal{D}_1 \vdash_R G_1, \mathcal{G} \qquad \mathcal{D}_2 \vdash_R G_2, \mathcal{G}'}{\mathcal{D}_1, \mathcal{D}_2 \vdash_R G_1 \otimes G_2, \mathcal{G}, \mathcal{G}'}$$

5. The $\oplus$-rule:

$$\frac{\mathcal{D} \vdash_R G_1, \mathcal{G}}{\mathcal{D} \vdash_R G_1 \oplus G_2, \mathcal{G}} \quad \frac{\mathcal{D} \vdash_R G_2, \mathcal{G}}{\mathcal{D} \vdash_R G_1 \oplus G_2, \mathcal{G}}$$

6. The $⅋$-rule:

$$\frac{\mathcal{D} \vdash_R G_1, G_2, \mathcal{G}}{\mathcal{D} \vdash_R G_1 ⅋ G_2, \mathcal{G}}$$

7. The &-rule:

$$\frac{\mathcal{D}' \vdash_R G_1', \mathcal{G}' \qquad \mathcal{D}'' \vdash_R G_2'', \mathcal{G}''}{\mathcal{D} \vdash_R G_1 \;\&\; G_2, \mathcal{G}}$$

where $\mathcal{D}'$ and $\mathcal{D}''$ are distinct copies of D, and where $G_1', \mathcal{G}'$ and $G_2'', \mathcal{G}''$ are obtained from $G_1, \mathcal{G}$ and $G_2, \mathcal{G}$ by the applying the renamings of $\mathcal{D}'$ and $\mathcal{D}''$ respectively;

8. The $\multimap$-rule:

$$\frac{\mathcal{D} \cup [D] \vdash_R G, \mathcal{G}}{\mathcal{D} \vdash_R D \multimap G, \mathcal{G}}$$

9. The $\bigwedge$-rule:

$$\frac{\mathcal{D} \vdash_R G[y/x], \mathcal{G}}{\mathcal{D} \vdash_R \bigwedge x . G, \mathcal{G}}$$

where y is not free in $\mathcal{D}$ or $\mathcal{G}$;

10. The $\bigvee$-rule:

$$\frac{\mathcal{D} \vdash_R G[t/x], \mathcal{G}}{\mathcal{D} \vdash_R \bigvee x . G, \mathcal{G}}$$

Note that Rules 1 and 2 affect other sequents in the proof, so that a resolution proof is well-defined only if all occurrences of Rules 1 and 2 yield compatible substitutions. This completes the definition of resolution proof. □

Rules 3-10 of Definition 4.4 can be considered to correspond to the right rules of uniform proofs and Rule 2 corresponds to the left rules, in the presence of the decomposition [–].

An important property of resolution proofs is the *instance property*; that is, if $D[\vec{t}/\vec{x}], \mathcal{P} \vdash_R \mathcal{G}$, then $D, \mathcal{P} \vdash_R \mathcal{G}$. Thus we may think of resolution proofs as those in which the only left rule required is the resolution rule, as the others have been encoded by the form of $[-]$. Hence we may concentrate on the right rules, and use only the resolution rule — a specialized form of $\multimap$-L — on atoms.

Before proceeding to prove the soundness of resolution proofs, we consider three small examples of a resolution proof. First, consider the construction, from the root to the leaves, of a resolution proof of the sequent

$$\bigwedge x\,.\,(p(x) \otimes (p(x) \multimap q(x))) \vdash q(t)\,.$$

We must construct a resolution proof of

$$[\bigwedge x\,.\,(p(x) \otimes (p(x) \multimap q(x)))] \vdash_R q(t)\,.$$

¿From our earlier example, we know that

$$[\bigwedge x\,.\,(p(x) \otimes (p(x) \multimap q(x)))] = \{\, p(x)\,,\, p(x) \multimap q(x)\,\}\,,$$

and so we must show that

$$\{\, p(x)\,,\, p(x) \multimap q(x)\,\} \vdash_R q(t)$$

has a resolution proof. We apply the resolution rule, selecting the clause $p(x) \multimap q(x)$, and obtain the subgoals

$$\{\, p(t)\,\} \vdash_R p(t) \quad \text{and} \quad \{\, q(t)\,\} \vdash_R q(t)\,.$$

Both of these are axioms, and the resolution proof is completed.

Now consider the sequent

$$\bigwedge x\,.\,(p(x) \otimes q(x)) \vdash p(t) \otimes q(u)\,,$$

which is not provable in linear sequent calculus. We attempt to construct a resolution proof of

$$[\bigwedge x\,.\,(p(x) \otimes q(x))] \vdash_R p(t) \otimes q(u)\,,$$

i.e., of

$$\{\, p(x), q(x)\,\} \vdash_R p(t) \otimes q(u)\,.$$

We apply the $\otimes$-rule, and obtain the subgoals

$$\{\, p(x)\,\} \vdash_R p(t) \quad \text{and} \quad \{\, q(x)\,\} \vdash_R q(u)\,.$$

Now, both of these are of the form of axioms, but they are not compatible as the first requires x to be replaced by t and the second requires x to replaced by u. Consequently, we have failed to obtain a resolution proof.

Now consider the sequent

$$q \multimap p \vdash q^{\perp}, p\,,$$

which is provable in linear sequent calculus. We apply the $\perp$-rule to obtain the subgoal

$$q, q \multimap p \vdash p\,.$$

By applying the resolution rule, we obtain the subgoals

$$\{q\} \vdash_R q \quad \text{and} \quad \{p\} \vdash_R p,$$

and both of these are axioms.

We now come to the soundness of resolution proofs.

PROPOSITION 4.5 (SOUNDNESS OF RESOLUTION PROOFS) *Let $\mathcal{P}$ be a multiset of definite formulae, and let $\mathcal{G}$ be a multiset of goal formulae. If $[\mathcal{P}] \vdash_R \mathcal{G}$, then $\mathcal{P} \vdash \mathcal{G}$ has a simple locally LR proof.*

PROOF

We proceed by induction on the height of the resolution proof. By Theorem 3.18 it will suffice to show that $\mathcal{P} \vdash \mathcal{G}$ has a linear proof.

The base case occurs when $[\mathcal{P}] = \{L\}$ for some positive literal L, and it is clear, by considering possibly several applications of the $\bigwedge$-L, $\otimes$-L and $\&$-L rules, that the proposition holds in this case.

Hence we assume that the proposition holds for all sequents whose resolution proof is no more than a given height.

Consider the step used to derive the current sequent. There are nine cases:

resolution: In this case we have that $[\mathcal{P}'][\vec{t}/\vec{x}] \vdash_R G', \mathcal{G}'$ where $[\mathcal{P}'] \cup \{G \multimap L\} = [\mathcal{P}]$, $G' \multimap L' = (G \multimap L)[\vec{t}/\vec{x}]$, and $\mathcal{G} = \mathcal{G}' \cup \{L'\}$. By the hypothesis, $\mathcal{P}'[\vec{t}/\vec{x}] \vdash G', \mathcal{G}'$ has a proof, and as $L' \vdash L'$ is an initial sequent, we may apply the $\multimap$-L rule to derive $\mathcal{P}'[\vec{t}/\vec{x}], G' \multimap L' \vdash \mathcal{G}', L'$. Then we may apply $\bigwedge$-L, $\&$-L and/or $\otimes$-L a number of times to get a proof of $\mathcal{P} \vdash \mathcal{G}$.

$\perp$-rule: In this case we have that $[\mathcal{P}], L \vdash_R \mathcal{G}'$ where $\mathcal{G} = \{L^{\perp}\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P}, L \vdash \mathcal{G}'$ has a simple locally LR proof, and so $\mathcal{P} \vdash L^{\perp}, \mathcal{G}'$ has a simple locally LR proof.

$\&$-rule: In this case we have that $\mathcal{P}_1 \vdash_R G_1', \mathcal{G}''$ and $\mathcal{P}_2 \vdash_R G_2', \mathcal{G}'''$ where $\mathcal{G} = \{G_1 \& G_2\} \cup \mathcal{G}'$, $\mathcal{P}_1$, $\mathcal{P}_2$ are distinct copies of $[\mathcal{P}]$ and G_1', G_2', $\mathcal{G}''$ and $\mathcal{G}'''$ are the correspondingly updated versions of G_1, G_2 and $\mathcal{G}'$ respectively, and so by the hypothesis $\mathcal{P}_1' \vdash G_1', \mathcal{G}''$ and $\mathcal{P}_2' \vdash G_2', \mathcal{G}'''$ are provable, where $\mathcal{P}_1'$ and $\mathcal{P}_2'$ are distinct copies of $\mathcal{P}$. It follows that $\mathcal{P} \vdash G_1, \mathcal{G}'$ and $\mathcal{P} \vdash G_2, \mathcal{G}'$ are provable, and so $\mathcal{P} \vdash G_1 \& G_2, \mathcal{G}'$ is provable.

$\otimes$-rule: In this case we have that $\mathcal{P}_1 \vdash_R G_1, \mathcal{G}_1$ and $\mathcal{P}_2 \vdash_R G_2, \mathcal{G}_2$ where $[\mathcal{P}] = \mathcal{P}_1 \cup \mathcal{P}_2$ and $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$, and so by the hypothesis $\mathcal{P}_1 \vdash G_1, \mathcal{G}_1$ and $\mathcal{P}_2 \vdash G_2, \mathcal{G}_2$ are provable, as $[\mathcal{P}_1] = \mathcal{P}_1$ and $[\mathcal{P}_2] = \mathcal{P}_2$, and so $\mathcal{P}_1, \mathcal{P}_2 \vdash G_1 \otimes G_2, \mathcal{G}$ is provable. We may then apply $\bigwedge$-L, $\&$-L and $\otimes$-L to the latter sequent to obtain a proof of $\mathcal{P} \vdash G_1 \otimes G_2, \mathcal{G}$.

$\oplus$-rule: In this case we have that $[\mathcal{P}] \vdash_R G_1, \mathcal{G}'$ or $[\mathcal{P}] \vdash_R G_2, \mathcal{G}'$ where $\mathcal{G} = \{G_1 \oplus G_2\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P} \vdash G_1, \mathcal{G}'$ or $\mathcal{P} \vdash G_2, \mathcal{G}'$ have simple locally LR proofs, and so $\mathcal{P} \vdash G_1 \oplus G_2, \mathcal{G}'$ has a simple locally LR proof.

$⅋$-rule: In this case we have that $[\mathcal{P}] \vdash_R G_1, G_2, \mathcal{G}'$ where $\mathcal{G} = \{G_1 ⅋ G_2\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P} \vdash G_1, G_2, \mathcal{G}'$ has a simple locally LR proof, and so $\mathcal{P} \vdash G_1 ⅋ G_2, \mathcal{G}'$ has a simple locally LR proof.

$\multimap$-rule: In this case we have that $[\mathcal{P}], [D] \vdash_R G, \mathcal{G}'$ where $\mathcal{G} = \{D \multimap G\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P}, D \vdash G, \mathcal{G}'$ has a simple locally LR proof, and so $\mathcal{P} \vdash D \multimap G, \mathcal{G}'$ has a simple locally LR proof.

$\bigwedge$-rule: In this case we have that $[\mathcal{P}] \vdash_R G[y/x], \mathcal{G}'$ where $\mathcal{G} = \{\bigwedge x.G\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P} \vdash G[y/x], \mathcal{G}'$ has a simple locally LR proof, and so $\mathcal{P} \vdash \bigwedge x . G, \mathcal{G}'$ has a simple locally LR proof.

$\bigvee$-rule: In this case we have that $[\mathcal{P}] \vdash_R G[t/x], \mathcal{G}'$ where $\mathcal{G} = \{\bigvee x . G\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P} \vdash G[y/x], \mathcal{G}'$ has a simple locally LR proof, and so $\mathcal{P} \vdash \bigvee x . G, \mathcal{G}'$ has a simple locally LR proof.

Hence, as $\mathcal{P} \vdash \mathcal{G}$, by Theorem 3.18, $\mathcal{P} \vdash \mathcal{G}$ has a simple locally LR proof. This completes the proof. □

Next comes the completeness of resolution proofs.

PROPOSITION 4.6 (COMPLETENESS OF RESOLUTION PROOFS) *Let $\mathcal{P}$ be a multiset of definite formulae, and let $\mathcal{G}$ be a multiset of goal formulae. If $\mathcal{P} \vdash \mathcal{G}$ has a simple locally LR proof, then $[\mathcal{P}] \vdash_R \mathcal{G}$.*

PROOF
We proceed by induction on the length of the proof of $\mathcal{P} \vdash \mathcal{G}$. The base case occurs when $\mathcal{P} \vdash \mathcal{G}$ is initial, *i.e.* $\mathcal{P} = \mathcal{G} = \{L\}$ for some positive literal L, and clearly the proposition holds in this case.

Hence we assume that the proposition holds for all sequents whose proof is of no more than a given height. Consider the rule used to derive $\mathcal{P} \vdash \mathcal{G}$. By Lemma 3.17, we need only consider the rules $\otimes$-L, &-L, $\bigwedge$-L, $\multimap$-L, $\bot$-R, $\otimes$-R, & -R, $⅋$-R, $\oplus$-R, $\multimap$-R, $\bigwedge$-R and $\bigvee$-R.

$\otimes$-L: The previous sequent must be $D_1, D_2, \mathcal{P}' \vdash \mathcal{G}$ where $\{D_1 \otimes D_2\} \cup \mathcal{P}' = \mathcal{P}$, and so by the hypothesis, $[D_1], [D_2], [\mathcal{P}'] \vdash_R \mathcal{G}$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

& -L: The previous sequent must be either $D_1, \mathcal{P}' \vdash \mathcal{G}$ or $D_2, \mathcal{P}' \vdash \mathcal{G}$ where $\{D_1 \,\&\, D_2\} \cup \mathcal{P}' = \mathcal{P}$, and so by the hypothesis, either $[D_1], [\mathcal{P}'] \vdash_R \mathcal{G}$ or $[D_2], [\mathcal{P}'] \vdash_R \mathcal{G}$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\bigwedge$-L: The previous sequent must be $D[t/x], \mathcal{P}' \vdash \mathcal{G}$, where $\{\bigwedge x . D\} \cup \mathcal{P}' = \mathcal{P}$, and so by the hypothesis we have that $[D[t/x], \mathcal{P}'] \vdash_R \mathcal{G}$ has a resolution proof, *i.e.*, $[D[t/x]] \cup [\mathcal{P}'] \vdash_R \mathcal{G}$ has a resolution proof. Now it follows from this and the instance property of resolution proofs that $[D] \cup [\mathcal{P}'] \vdash_R \mathcal{G}$ has a resolution proof — the resolution proof of the induction hypothesis will suffice — and so $[\bigwedge x . D, \mathcal{P}'] \vdash_R \mathcal{G}$.

$\multimap$-L: The previous sequent must be $\mathcal{P}' \vdash G, \mathcal{G}'$ where $\{G \multimap L\} \cup \mathcal{P}' = \mathcal{P}$ and $\{L\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}'] \vdash_R G, \mathcal{G}'$, and by an application of the resolution rule $[\mathcal{P}'], G \multimap L \vdash_R L, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\perp$-R: The previous sequent must be $\mathcal{P}, L \vdash \mathcal{G}'$, where $\{L^\perp\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}], L \vdash_R G, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\otimes$-R: The previous sequents must be $\mathcal{P}_1 \vdash G_1, \mathcal{G}_1$ and $\mathcal{P}_2 \vdash G_2, \mathcal{G}_2$, where $\mathcal{P}_1 \cup \mathcal{P}_2 = \mathcal{P}$ and $\{G_1 \otimes G_2\}, \mathcal{G}_1 \cup \mathcal{G}_2 = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}_1] \vdash_R G_1, \mathcal{G}_1$ and $[\mathcal{P}_2] \vdash_R G_2, \mathcal{G}_2$, and so $[\mathcal{P}_1], [\mathcal{P}_2] \vdash_R G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

& -R: The previous sequents must be $\mathcal{P} \vdash G_1, \mathcal{G}'$ and $\mathcal{P} \vdash G_2, \mathcal{G}'$, where $\{G_1 \,\&\, G_2\} \cup \mathcal{G}' = \mathcal{G}$, and so the sequents $\mathcal{P}' \vdash G_1', \mathcal{G}''$ and $\mathcal{P}'' \vdash G_2', \mathcal{G}'''$ are provable, where $\mathcal{P}'$ and $\mathcal{P}''$ are distinct copies of P and G_1', G_2', $\mathcal{G}''$ and $\mathcal{G}'''$ are the correspondingly updates versions of G_1, G_2 and $\mathcal{G}'$ respectively, and so by the hypothesis, $[\mathcal{P}'] \vdash_R G_1', \mathcal{G}''$ and $[\mathcal{P}''] \vdash_R G_2', \mathcal{G}'''$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\oplus$-R: The previous sequent must be either $\mathcal{P} \vdash G_1, \mathcal{G}'$ or $\mathcal{P} \vdash G_2, \mathcal{G}'$, where $\{G_1 \oplus G_2\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}] \vdash_R G_1, \mathcal{G}'$ or $[\mathcal{P}] \vdash_R G_2, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$⅋$ -R: The previous sequent must be $\mathcal{P} \vdash G_1, G_2, \mathcal{G}'$, where $\{G_1 ⅋ G_2\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}] \vdash_R G_1, G_2, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\multimap$ -R: The previous sequent must be $\mathcal{P}, D \vdash G, \mathcal{G}'$, where $\{D \multimap G\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}], [D] \vdash_R G, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\bigwedge$-R: The previous sequent must be $\mathcal{P} \vdash G[y/x], \mathcal{G}'$, where $\{\bigwedge x \,.\, G\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}] \vdash_R G[y/x], \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\bigvee$-R: The previous sequent must be $\mathcal{P} \vdash G[t/x], \mathcal{G}'$, where $\{\bigvee x \,.\, G\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}] \vdash_R G[t/x], \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

This completes the proof. □

Thus we arrive at the following theorem:

THEOREM 4.7 *Let $\mathcal{P}$ be a multiset of definite formulae and let $\mathcal{G}$ be a multiset of goal formulae. Then $\mathcal{P} \vdash \mathcal{G}$ is provable iff $[\mathcal{P}] \vdash_R \mathcal{G}$.*

PROOF
The proof follows immediately from Corollary 3.18 and Propositions 4.5 and 4.6. □

In this way we may think of of resolution proofs a relatively efficient way of proving theorems in the given fragment of linear logic. Furthermore, such proofs form a basis for logic programming in this fragment, as the choice of step in the proof-search process is determined by the structure of the goal[15], and so any (multiset of) goal formulae may be considered as a goal.

[15]The search steps are not totally determined in linear logic. Some choice remains in the decomposition of antecedents and succedents in the $\otimes$-R rule. One could simply exclude $\otimes$ from goal formulae. However we propose, in work in progress on an implementation of our notion of linear logic programming, to handle such choices by certain techniques at the level of an abstract interpreter and mechanical implementation. This is point discussed in some detail in Section 7.

5 A "Herbrand" Quantale Semantics

The semantics of logic programming in both classical and intuitionistic logic is usually given in terms of a *least fixed point* construction over a collection of *Herbrand interpretations* [18], [19]. These constructions rely upon the monotonicity of the consequence relation of operational proof.[16]

For example, Miller [19] provides a least fixed point construction of a Kripke-like model of intuitionistic logic programs (hereditary Harrop formulae). The main point here is that program clauses can be reused; so that, informally, if the program $\mathcal{P}$ can compute the goal G after a finite number of computation steps then the program $\mathcal{P}'$, where $\mathcal{P} \subseteq \mathcal{P}'$, can compute the goal G after a finite number of computation steps.

In linear logic such a monotonicity property is not available. However, we can construct a semantics for linear logic programming by considering the *quantale semantics* of linear logic.

The reader is referred to Appendix B for the basic definitions of quantales and for the standard soundness and completeness results for the interpretation of linear logic in quantales [24]. In this section we construct a family of interpretations, in the *linear term quantale* of Appendix B, one for each program $\mathcal{P}$.

We do this by "parameterizing" the interpretation $| - |_{\mathbf{P(M)}}$ of Appendix B to give an interpretation $\mathrm{H}_{\mathcal{P}}$ such that $\mathrm{H}_{\mathcal{P}}(\mathcal{G}) = \mathrm{Pr}_{\mathcal{P}}(\mathcal{G})$, where

$$\mathrm{Pr}_{\mathcal{P}}(\mathcal{G}) =_{\mathrm{def}} \{\mathcal{B} \mid \mathcal{P} \vdash \mathcal{G}\, , \mathcal{B} \text{ is provable}\},$$

in which we interpret $\mathcal{B} \equiv B_1, \ldots, B_n$ as the formula $B_1 \maltese \ldots \maltese B_n$ and in which "provable" means provable in linear sequent calculus.

THEOREM 5.1 (HERBRAND COMPLETENESS) *If* $\mathrm{H}_{\mathcal{P}}(\mathcal{G})$ *is valid then* $[\mathcal{P}] \vdash_R \mathcal{G}$ *has a resolution proof.*

PROOF SKETCH

By Theorem 4.7 it is sufficient to prove that if the goal $\mathcal{G}$ is valid in $\mathrm{H}_{\mathcal{P}}$ then $\mathcal{P} \vdash \mathcal{G}$ has a proof in linear sequent calculus. We demonstrate the existence, for each program $\mathcal{P}$, of an interpretation $\mathrm{H}_{\mathcal{P}}$ such that

$$\mathrm{H}_{\mathcal{P}}(\mathcal{G}) = \mathrm{Pr}_{\mathcal{P}}(\mathcal{G})\,.$$

We begin by defining

$$\mathrm{H}_{\mathcal{P}}(A) =_{\mathrm{def}} \mathrm{Pr}_{\mathcal{P}}(A)$$

for each atomic formula $A \in \mathcal{A}(\mathcal{L})$. By induction on the structure of $\mathcal{G}$, and by utilizing the definition of the linear connectives in the term quantale, we prove that $\mathrm{H}_{\mathcal{P}}(\mathcal{G}) = \mathrm{Pr}_{\mathcal{P}}(\mathcal{G})$. This part of the proof can be copied almost exactly from the proof of the completeness

[16]The reader is referred to recent work of Avron [5], [6] on consequence relations, which pays particular attention to sequent calculi.

theorem of [14] (see also Appendix B), provided one takes care of the parameterization of our construction by programs $\mathcal{P}$.

Finally, we prove a lemma to the effect that $\mathbf{1} \leq \mathrm{Pr}_{\mathcal{P}}(\mathcal{G})$ if and only if $\mathcal{P} \vdash \mathcal{G}$ is provable. This part of the proof can be copied almost exactly from the proof of the completeness theorem of [25].

The required result follows. □

We give some examples of the Herbrand quantale semantics:

1. Let $\mathcal{P}_1$ be the program that consists of the single atomic definite formula p and let q be an atomic goal formula; then

$$\begin{array}{rcl} \mathrm{H}_{\mathcal{P}_1}(q) & = & \{\, \mathcal{B} \mid p \vdash q, \mathcal{B} \,\} \\ & = & \{\, (p \multimap q)^{\perp}, r^{\perp} \maltese (r \multimap (p \multimap q))^{\perp}, \ldots \} \,. \end{array}$$

   We remark that it is a simple matter to verify that the sequents

$$p \vdash q \,,\, (p \multimap q)^{\perp} \ \text{ and } \ p \vdash q \,,\, r^{\perp} \maltese (r \multimap (p \multimap q))^{\perp}$$

   are provable in linear sequent calculus.

2. Let $\mathcal{P}_2$ be the program that consists of the single definite formula $p \multimap q$; then

$$\begin{array}{rcl} \mathrm{H}_{\mathcal{P}_2}(q) & = & \{\, \mathcal{B} \mid p \multimap q \vdash q, \mathcal{B} \,\} \\ & = & \{\, p^{\perp}, (r \multimap p)^{\perp} \maltese r^{\perp}, \ldots \} \,. \end{array}$$

   We remark that it is a simple matter to verify that the sequents

$$p \multimap q \vdash q \,,\, p^{\perp} \ \text{ and } \ p \multimap q \vdash q \,,\, (r \multimap p)^{\perp} \maltese r^{\perp}$$

   are provable in linear sequent calculus.

3. Let $\mathcal{P}_3$ be the program that consists of the single definite formula $\bigwedge x \,.\, (p(x) \multimap q(x))$ and let $q(t)$ be an atomic formula; then

$$\begin{array}{rcl} \mathrm{H}_{\mathcal{P}_3}(q(t)) & = & \{\, \mathcal{B} \mid \bigwedge x \,.\, p(x) \multimap q(x) \vdash q(t), \mathcal{B} \,\} \\ & = & \{\, p(t)^{\perp}, (r(t) \multimap p(t))^{\perp} \maltese r(t)^{\perp}, \ldots \} \,. \end{array}$$

   We remark that it is a simple matter to verify that the sequents

$$\bigwedge x \,.\, p(x) \multimap q(x) \vdash q(t) \,,\, p^{\perp} \ \text{ and } \ \bigwedge x \,.\, p(x) \multimap q(x) \vdash q(t) \,,\, (r(t) \multimap p(t))^{\perp} \maltese r(t)^{\perp}$$

   are provable in linear sequent calculus.

We remark that this semantics is inadequate in several ways. For example, it fails to provide what we might think of as a *clause-directed denotational semantics* in the manner of the T-operators of intuitionistic logics.[17]

We plan, in further work, to construct a categorical semantics for linear logic programming, perhaps in the manner of [4].

[17]See [20] and [22] for a discussion of this in the setting of intuitionistic hereditary Harrop formulae and the $\lambda\Pi$-calculus respectively.

6 Intuitionistic Logic

We have seen that one of the main difference between intuitionistic logic and linear logic in the context of logic programming is that the resolution rule in the latter case requires the clause used to be deleted from the program. This difference is essentially due to the lack of the weakening rule in linear logic. However somewhat restricted forms of contraction and weakening are present in linear logic through the exponentials ! and ?. Hence it seems natural to encode intuitionistic logic in linear logic by means of these connectives. In this section we show how this may be done — essentially by placing the exponential ! in front of the clauses, and defining an extra resolution rule for this case.

First we consider the larger class of definite and goal formulae given below.

DEFINITION 6.1 (EXTENDED DEFINITE FORMULAE AND EXTENDED GOAL FORMULAE) *Let A range over atomic formulae. We define the classes of* positive literals, extended definite formulae *and* extended goal formulae *as follows:*

$$\begin{array}{lcll}
\textit{Positive Literals} & L & ::= & \mathbf{0} \mid \mathbf{1} \mid \bot \mid \top \mid A \\
\textit{Extended definite formulae} & D & ::= & L \mid D \,\&\, D \mid D \otimes D \mid G \multimap L \mid \bigwedge x \,.\, D \\
 & & & \mid\, !D \\
\textit{Extended goal formulae} & G & ::= & L \mid L^{\perp} \mid G \otimes G \mid G \oplus G \mid G \parr G \mid G \,\&\, G \\
 & & & \mid D \multimap G \mid \bigwedge x \,.\, G \mid \bigvee x \,.\, G
\end{array}$$

This completes the extension of definite formulae and goal formulae. □

We use the prefix "extended" in order to disambiguate between the latter classes of formulae and those defined in Section 2. When it is clear from the context which class is meant, we will often omit the word extended.

It is easy to see that a lemma analogous to Lemma 3.17 holds.

LEMMA 6.2 *Let $\mathcal{P}$ be a program and $\mathcal{G}$ be a goal. A proof of $\mathcal{P} \vdash \mathcal{G}$ contains no occurrences of $\bot$-L, $\parr$-L, $\oplus$-L, ?-L, or $\bigvee$-L.*

PROOF
A simple induction on the structure of proofs. □

Now this result together with the permutation results of Section 3 allow us to derive the following proposition.

PROPOSITION 6.3 *Let $\mathcal{P}$ be a multiset of extended definite formulae and let $\mathcal{G}$ be a multiset of extended goal formulae. Then $\mathcal{P} \vdash \mathcal{G}$ has a linear proof iff $\mathcal{P} \vdash \mathcal{G}$ has a simple locally LR proof.*

PROOF
By Lemma 6.2 we have that any linear proof of $\Gamma \vdash \Delta$ does not contain $\bot$-L, $\parr$-L, $\oplus$-L, ?-L or $\bigvee$-L; and so by Corollary 3.13, $\Gamma \vdash \Delta$ has a locally LR proof. Hence by Proposition 3.16 $\Gamma \vdash \Delta$ has a simple locally LR proof. □

Thus we only need to concentrate on the extensions to $[D]$ and $\vdash_R$. First we extend the notion of a clause to include the exponential !.

DEFINITION 6.4 (CLAUSE) *A* linear clause *is a formula of the form L or $G \multimap L$, where L ranges over positive literals and G ranges over goal formulae. A* mixed clause *is a formula of the form L, $G \multimap L$ or $!(C_1 \otimes \ldots \otimes C_n)$, where each C_i, $1 \leq i \leq n$, is a linear clause. We let C range over mixed clauses.* □

Note that when G is a conjunction of atoms, a linear clause in the above sense is just a *Horn* clause.

We will find it convenient to divide an antecedent into two parts — those formulae which are of the form $!F$, and those which are not.

DEFINITION 6.5 (ANTECEDENT DIVISION) *Let $\cup$ denote multiset union, and let $\mathcal{D}$ be a multiset of formulae. We define two multisets $\mathcal{D}^I$ and $\mathcal{D}^L$ such that $\mathcal{D} = \mathcal{D}^I \cup \mathcal{D}^L$ as follows:*

$$\mathcal{D}^I = \bigcup_{F \in \mathcal{D}} \{\ F \mid \textit{the outermost connective of } F \textit{ is } !\ \}$$

$$\mathcal{D}^L = \bigcup_{F \in \mathcal{D}} \{\ F \mid \textit{the outermost connective of } F \textit{ is not } !\ \} \ \square$$

We may think of this distinction as specifying which formulae are known to be able to be re-used (those commencing with !) and those which are not.

We also need to extend the mapping $[-]$, so that we can generate a multiset of mixed clauses from an extended program.

DEFINITION 6.6 (EXTENDED CLAUSAL DECOMPOSITION) *Let $\cup$ denote multiset union. We define a mapping $[-]$ from definite formulae to multisets of definite formulae as follows:*

$$\begin{array}{lll}
[\mathcal{P}] & =_{\text{def}} & \bigcup_{D \in \mathcal{P}} [D] \\
[L] & =_{\text{def}} & \{ L \} \\
[D_1 \,\&\, D_2] & =_{\text{def}} & [D_1] \text{ or } [D_2] \\
[D_1 \otimes D_2] & =_{\text{def}} & [D_1] \cup [D_2] \\
[G \multimap L] & =_{\text{def}} & \{ G \multimap L \} \\
[\bigwedge x \,.\, D] & =_{\text{def}} & [D] \\
[!\,D] & =_{\text{def}} & \{ !\ \bigotimes_{C \in [D]^L} C \} \cup [D]^I
\end{array}$$

where variables x that occur in $\mathcal{P}$ and outwith *the scope of any ! are marked as* global variables *and where variables x that occur in $\mathcal{P}$* within *the scope of some ! are marked as* local. □

Note that $[\mathcal{P}]$ is a multiset of mixed clauses. Note also that there are free variables in the clauses in $[\mathcal{P}]$, with certain of the free variables marked as being "global" and the other free variables marked as being "local"; these are the variables that are "outermost" in $\mathcal{P}$. One point to note is that the mixture of universal quantifiers and ! may lead to some (possibly) surprising results. For example, let $\mathcal{P}$ be the single definite formula $\bigwedge x \,.\, ! \bigwedge y \,.\, p(x, y)$, and consider the goal $p(a,b) \otimes p(a,c)$. It should be clear that $! \bigwedge y . p(a,y) \vdash p(a,b) \otimes p(a,c)$ is provable, and so $\bigwedge x . ! \bigwedge y . p(x,y) \vdash p(a,b) \otimes p(a,c)$ is provable. However $\bigwedge x . ! \bigwedge y . p(x,y) \vdash p(a,b) \otimes p(b,c)$ is not provable. Hence we may think of universally quantified variables outside the scope of ! as "global", in that all occurrences of the variable must be updated consistently. Due to the possibility of splitting the program during computation, this may mean updating variables simultaneously across several branches of the proof.[18] On the other hand, the universally quantified variables within the scope of ! may be thought of as "local", in that they need only be updated in one formula. This leads us to the definitions that follow.

DEFINITION 6.7 (GLOBAL AND LOCAL VARIABLES) *Let $\mathcal{P}$ be a program. If x is a free variable in $[\mathcal{P}]$, then it is a* global variable *if it occurs outside the scope of any ! in P. Otherwise, x is a* local variable. □

Note that the marked "global" and "local" variables of Definition 6.6 satisfy this definition. For a given substitution $[\vec{t}/\vec{x}]$, we write $[\vec{t}/\vec{x}]^g$ to denote that the application of the substitution is to only those free variables that are global, throughout the proof. Otherwise, $[\vec{t}/\vec{x}]$ applies to all global variables, throughout the proof, and to local variables only in the sequent in which the substitution arises (and to those above it, of course). Clearly one way to enforce this discipline is to ensure that local variables are unique to each sequent, so that local variables in different sequents have different names.

One of the major differences between intuitionistic logic programming and linear logic programming is that in the latter case we may need to split the program (and indeed the goal), due to the form of $\otimes$-R rule. In the case of linear clauses, this is simply a matter of dividing up the given multiset. However for mixed clauses, we need a more sophisticated approach, as formulae beginning with a ! may be used any number of times in a proof. From the contraction rule we know that if $\mathcal{P}, !D, !D \vdash \mathcal{G}$ is provable, then so is $\mathcal{P}, !D \vdash \mathcal{G}$, and so when searching for an approprate splitting of the program, we need to add formula of the form $!D$ to each branch. We will also need to "balance" the use of subformulae of $!D$, as we can only add an arbitrary number of copies of D to the antecedent, and not arbitrary numbers of a given subformula of D. For example, consider the program $!(p \otimes q)$ and the goal $p \otimes (q \otimes q)$.[19] By use of the contraction rule in conjunction with !-L and $\otimes$-L, the program will imply the goal if $p, q, !(p \otimes q) \vdash p \otimes (q \otimes q)$ is provable. If we reduce this to $p, !(p \otimes q) \vdash p$ and $q, !(p \otimes q) \vdash q \otimes q$, then the first sequent is provable, but the second is not, as we need to add p, q to the antecedent before splitting again, and we find that $p, q, !(p \otimes q) \vdash q$ is not provable. However, if we were allowed to duplicate more copies of the subformula q of $p \otimes q$ than of the subformula p we should be able to prove

[18]In terms of the *extended resolution proofs* defined below, those branches that are above the same premiss of the last & –rule (Rule 7 in the definition of resolution proof, below) as the sequent to which the resolution rule is applied. The form of the &-rule in such proofs enforces the restriction of substitutions to their own &-branches.

[19]Note that the sequents $!(p \otimes q) \vdash p \otimes (q \otimes q)$, $!(p \otimes q) \vdash (p \otimes p) \otimes (q \otimes q \otimes q)$, *etc.*, are not provable in linear sequent calculus.

the goal $p \otimes (q \otimes q)$ from the program $!(p \otimes q)$. Hence we need to ensure that the splitting of the program is done in such as way as to respect the structure of the original formulae.

The reader should note however that as yet we have not provided any mechanism for the calculation of an appropriate splitting of the program (and goal). We postpone the description of such a procedure until we have defined the notion of resolution proof and are ready to describe an interpreter.

DEFINITION 6.8 (MULTISET EXPANSION) *Let $\mathcal{C}$ be a multiset and let $\cup$ denote multiset union. We define $\mathcal{C}^n$ for integers $n \geq 0$ inductively as follows:*

$$\mathcal{C}^0 = \emptyset \qquad \mathcal{C}^{n+1} = \mathcal{C} \cup \mathcal{C}^n \qquad \square$$

DEFINITION 6.9 (EXPANSION) *Let $\mathcal{D}$ be a multiset of mixed clauses, and suppose that $\mathcal{D}^I = \{!(C_{11} \otimes \ldots \otimes C_{1n_1}), \ldots, !(C_{m1} \otimes \ldots \otimes C_{mn_m})\}$. An* expansion *of $\mathcal{D}$ is a pair of multisets $\mathcal{D}_1$ and $\mathcal{D}_2$ such that*

$$\begin{array}{rcl} \mathcal{D}_1 \cup \mathcal{D}_2 & = & \mathcal{D}^L \ \cup \{ C_{11}, \ldots, C_{1n_1} \}^{i_1} \\ & & \vdots \\ & & \cup \{ C_{m1}, \ldots, C_{mn_m} \}^{i_m} \end{array}$$

for integers $i_1, \ldots, i_m \geq 0$. A linear expansion *of $\mathcal{D}$ is a pair of multisets $\mathcal{D}_1$ and $\mathcal{D}_2$ such that $\mathcal{D}_1 \cup \mathcal{D}_2 = \mathcal{D}^L$.* $\square$

Note that a linear expansion of $\mathcal{D}$ corresponds to the case in which $i_1 = \ldots = i_m = 0$.

We come now to the definition of (extended) resolution proof.

DEFINITION 6.10 (EXTENDED RESOLUTION PROOF) *We define the notion of extended resolution proof by defining a relation $\vdash_R$. Let $\mathcal{D}$ range over multisets of definite formulae and let $\cup$ denote multiset union. An* extended resolution proof *is tree regulated by the following rules, which is constructed from root to leaves, and is such that the resolution rules are applied only when no other rule is applicable:*

1. *The axiom judgement is given by:*

$$\frac{}{\mathcal{D} \vdash_R L}$$

where either $\mathcal{D}^L = \{ L' \}$ and $L = L'[\vec{t}/\vec{x}]$, or $\mathcal{D}^L = \emptyset$ and there exists $!L' \in \mathcal{D}^I$ such that $L = L'[\vec{t}/\vec{x}]$;

2. *We shall refer to this rule as the* resolution *rule:*

$$\frac{\mathcal{D}[\vec{t}/\vec{x}]^g \vdash_R G, \mathcal{G} \qquad \{ L \} \vdash_R L}{\mathcal{D} \cup \{G' \multimap L'\} \vdash_R L, \mathcal{G}}$$

where $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$;

3. *We shall refer to this rule as the* !-resolution *rule:*

$$\frac{\mathcal{D}[\vec{t}/\vec{x}]^g \cup [\vec{C}[\vec{t}/\vec{x}]] \cup \{!((G' \multimap L') \otimes \vec{C})[\vec{t}/\vec{x}]^g\} \vdash_R G\,,\,\mathcal{G} \qquad \{\,L\,\} \vdash_R L}{\mathcal{D} \cup \{!((G' \multimap L') \otimes \vec{C})\} \vdash_R L, \mathcal{G}}$$

where $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$*, and where* $\vec{C}$ *denotes* $C_1 \otimes \ldots \otimes C_m$ *for some* C_i*s,* $1 \leq i \leq m$ [20]*;*

4. *The* $\perp$*-rule:*

$$\frac{\mathcal{D} \cup \{\,L\,\} \vdash_R \mathcal{G}}{\mathcal{D} \vdash_R L^{\perp}, \mathcal{G}}$$

5. *The* $\otimes$*-rule:*

$$\frac{\mathcal{D}^I, \mathcal{D}_1^I, \mathcal{D}_1 \vdash_R G_1, \mathcal{G} \qquad \mathcal{D}^I, \mathcal{D}_2^I, \mathcal{D}_2 \vdash_R G_2, \mathcal{G}'}{\mathcal{D} \vdash_R G_1 \otimes G_2, \mathcal{G}, \mathcal{G}'}$$

where $\mathcal{D}_1$*,* $\mathcal{D}_2$ *is a linear expansion of* $\mathcal{D}$*, and where* $\mathcal{D}_1 \cup \mathcal{D}_1^I$*,* $\mathcal{D}_2 \cup \mathcal{D}_2^I$ *is an expansion of* $\mathcal{D}$*;*

6. *The* $\oplus$*-rule:*

$$\frac{\mathcal{D} \vdash_R G_1, \mathcal{G}}{\mathcal{D} \vdash_R G_1 \oplus G_2, \mathcal{G}} \qquad \frac{\mathcal{D} \vdash_R G_2, \mathcal{G}}{\mathcal{D} \vdash_R G_1 \oplus G_2, \mathcal{G}}$$

7. *The* $⅋$*-rule:*

$$\frac{\mathcal{D} \vdash_R G_1, G_2, \mathcal{G}}{\mathcal{D} \vdash_R G_1 ⅋ G_2, \mathcal{G}}$$

8. *The* &*-rule:*

$$\frac{\mathcal{D}' \vdash_R G_1', \mathcal{G}' \qquad \mathcal{D}'' \vdash_R G_2'', \mathcal{G}''}{\mathcal{D} \vdash_R G_1 \,\&\, G_2, \mathcal{G}}$$

where $\mathcal{D}'$ *and* $\mathcal{D}''$ *are distinct copies of* D*, and where* $G_1', \mathcal{G}'$ *and* $G_2'', \mathcal{G}''$ *are obtained from* $G_1, \mathcal{G}$ *and* $G_2, \mathcal{G}$ *by the applying the renamings of* $\mathcal{D}'$ *and* $\mathcal{D}''$ *respectively. Note that we maintain "global" and "local" markings;*

9. *The* $\multimap$*-rule:*

$$\frac{\mathcal{D} \cup [D] \vdash_R G, \mathcal{G}}{\mathcal{D} \vdash_R D \multimap G, \mathcal{G}}$$

10. *The* $\bigwedge$*-rule:*

$$\frac{\mathcal{D} \vdash_R G[y/x], \mathcal{G}}{\mathcal{D} \vdash_R \bigwedge x\,.\,G, \mathcal{G}}$$

where y *is not free in* $\mathcal{D}$ *or* $\mathcal{G}$*.*

[20]Note that there may be no such component, *i.e.*, that m may be 0.

11. The $\bigvee$-rule:

$$\frac{\mathcal{D} \vdash_R G[t/x], \mathcal{G}}{\mathcal{D} \vdash_R \bigvee x \,.\, G, \mathcal{G}}$$

Note that Rules 1, 2 and 3 affect other sequents in the proof, so that a resolution proof is well-defined only if all occurrences of Rules 1, 2 and 3 yield compatible substitutions. □ .

Note that the extended form of the axiom judgement is not an initial sequent. For example, from the above we immediately get that $!\phi, !\psi, \chi \vdash_R \chi$, and it is clear that this sequent may be derived from the initial sequent $\chi \vdash \chi$ via two applications of !-L.

It should be clear that the !-resolution rule and the second part of the axiom judgement are directly analogous to the corresponding rules in intuitionistic logic. The !-resolution rule clearly has the property that $\mathcal{P} \vdash_R A$ if there is a clause $!G' \multimap L'$ in $\mathcal{P}$ such that $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$ and $\mathcal{P} \vdash_R G$, which is a more intricate form of the resolution rule given in [20]. Similar remarks apply to the second part of the axiom judgement.

The only other significant difference is the form of the $\otimes$ rule. In the previous case, there were no occurrences of ! anywhere, and hence the C?-L rule would never be applicable to sequents consisting of programs and goals. However this does not apply for extended programs and extended goals. We saw in Section 3 how the permutation results for $\otimes$-R are somewhat more delicate than those for the other rules, particularly for the cases $\otimes$-L and C?-L. In the former case, we were able to circumvent the problem by the use of $[\mathcal{P}]$ rather than $\mathcal{P}$ itself. In this way we may think of $[-]$ as a compile-time transformation, so that $\mathcal{P}$ is transformed before execution into a form which is more easily manipulated. In particular, this means that no occurrences of $\otimes$-L are used in the search for a proof of the goal. In the latter case, we cannot get around the difficulty quite so easily, as we do not know in advance (*i.e.* at compile time) how many occurrences of C?-L may be needed, or equivalently how many times a given formula $!F$ will be needed in the proof. Hence, we need to preserve formulae of the form $!F$ in all parts of the proof where they may be needed. In particular, we need to include all such formulae in each antecedent created by the use of the $\otimes$-R as a search operation. Clearly if some or all of these formulae are not needed in the proof, they may be discarded (by the use of weakening as a search operator), and so no proofs will be missed. In this way the ! connective at the front of a formula ensures that it behaves in a fashion characteristic of intuitionistic logic, rather than of linear logic, and hence extended resolution proofs may be thought of as a mixture of purely linear deduction and intuitionistic deduction.

Before proceeding to develop the theory of extended resolution proofs, we give two small examples of the use of the !-resolution rule:

- Consider the sequent

$$\{\, q(x,z), !(((q(x,y) \otimes r(x,y)) \multimap p(x,y)) \otimes r(x,y)) \,\} \vdash_R p(t,u)\,.$$

By applying the !-resolution rule, we obtain the subgoals

$$\{\, p(t,u) \,\} \vdash_R p(t,u)$$

and

$$\{\, q(t,z), !(((q(t,y) \otimes r(t,y)) \multimap p(t,y)) \otimes r(t,y)), r(t,u) \,\} \vdash_R q(t,u) \otimes r(t,u)\,;$$

- Consider the sequent
$$\{\, q(a), !(q(a) \multimap p(a)) \vdash_R p(a)\,.$$
By applying the !-resolution rule, we obtain the subgoals
$$\{\, p(a) \,\} \vdash_R p(a) \text{ and } \{\, q(a), !(q(a) \multimap p(a)) \vdash_R p(a) \,\}\,.$$
This example corresponds to the usual resolution rule intuitionistic logic [20], [21].

We now proceed to show that extended resolution proofs behave in the expected manner. We show that the appropriate notions weakening and contraction are admissible rules.

LEMMA 6.11 (WEAKENING) *Let $\mathcal{P}$ be a multiset of mixed clauses, $!D$ be a mixed clause and let $\mathcal{G}$ be a multiset of goal formulae. If $\mathcal{P} \vdash_R \mathcal{G}$, then $\mathcal{P}, !D \vdash_R \mathcal{G}$.*

PROOF
We proceed by induction on the height of the proof. The base case occurs when $\mathcal{G}$ is just a positive literal L, and it is clear that the lemma holds in this case. Hence we assume that the lemma is true for all sequents whose proof is no more than a given height.

Consider the step used to derive the current sequent. There are ten cases.

resolution: The previous sequent must be $\mathcal{P}'[\vec{t}/\vec{x}] \vdash G, \mathcal{G}'$ where $\mathcal{P} = \mathcal{P}' \cup \{G' \multimap L'\}$, $\mathcal{G} = G, \mathcal{G}'$ and $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$. By the hypothesis we have $\mathcal{P}'[\vec{t}/\vec{x}], !D \vdash G, \mathcal{G}'$, and so we may apply the resolution rule to obtain $\mathcal{P}, !D \vdash_R \mathcal{G}$.

!-resolution: The previous sequent must be $\mathcal{P}[\vec{t}/\vec{x}]^g \cup [\vec{C}[\vec{t}/\vec{x}]] \cup \{!((G' \multimap L') \otimes \vec{C})[\vec{t}/\vec{x}]^g\} \vdash G, \mathcal{G}'$ where $\mathcal{P} = \mathcal{P}' \cup \{!((G' \multimap L') \otimes C)\}$, $\mathcal{G} = G, \mathcal{G}'$ and $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$. By the hypothesis we have $\mathcal{P}[\vec{t}/\vec{x}]^g \cup [\vec{C}[\vec{t}/\vec{x}]] \cup \{(!(G' \multimap L') \otimes \vec{C})[\vec{t}/\vec{x}]^g\}, !D \vdash G, \mathcal{G}'$ so we may apply the !-resolution rule to obtain $\mathcal{P}, !D \vdash_R \mathcal{G}$.

$\perp$-rule: The previous sequent must be $\mathcal{P}, L \vdash \mathcal{G}'$, where $\mathcal{G} = L^\perp, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D, L \vdash_R \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R L^\perp, \mathcal{G}'$.

& -rule: The previous sequents must be $\mathcal{P}' \vdash G_1', \mathcal{G}_1'$ and $\mathcal{P}'' \vdash G_2', \mathcal{G}_2'$, where $\mathcal{G} = G_1 \& G_2, \mathcal{G}_1, \mathcal{G}_2$, $\mathcal{P}'$, $\mathcal{P}''$ are distinct copies of $\mathcal{P}$ and G_1', $\mathcal{G}_1'$, G_2', $\mathcal{G}_2'$ are the correspondingly updated versions of G_1, $\mathcal{G}_1$, G_2, $\mathcal{G}_2$ respectively, and so by the hypothesis we have $\mathcal{P}, !D' \vdash_R G_1', \mathcal{G}_1'$ and $\mathcal{P}, !D'' \vdash_R G_2', \mathcal{G}_2'$ where $!D'$ and $!D''$ are distinct copies of $!D$, and so $\mathcal{P}, !D \vdash_R G_1 \& G_2, \mathcal{G}_1, \mathcal{G}_2$.

$\otimes$-rule: The previous sequents must be $\mathcal{P}^I, \mathcal{P}_1 \vdash G_1, \mathcal{G}_1$ and $\mathcal{P}^I, \mathcal{P}_2 \vdash G_2, \mathcal{G}_2$, where $\mathcal{P}_1$, $\mathcal{P}_2$ is an expansion of $\mathcal{P}$ and $\mathcal{G} = G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$, and so by the hypothesis we have $\mathcal{P}^I, \mathcal{P}_1, !D \vdash_R G_1, \mathcal{G}_1$ and $\mathcal{P}^I, \mathcal{P}_2, !D \vdash_R G_2, \mathcal{G}_2$, and so $\mathcal{P}^I, !D, \mathcal{P}_1, \mathcal{P}_2 \vdash_R G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$, *i.e.* $\mathcal{P}, !D \vdash_R G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$.

$\oplus$-rule: The previous sequent must be either $\mathcal{P} \vdash G_1, \mathcal{G}'$ or $\mathcal{P}_2 \vdash G_2, \mathcal{G}'$, where $\mathcal{G} = G_1 \oplus G_2, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}_1, !D \vdash_R G_1, \mathcal{G}'$ or $\mathcal{P}, !D \vdash_R G_2, \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R G_1 \oplus G_2, \mathcal{G}'$.

$\parr$-rule: The previous sequent must be $\mathcal{P} \vdash G_1, G_2, \mathcal{G}'$, where $\mathcal{G} = G_1 \parr G_2, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D \vdash_R G_1, G_2, \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R G_1 \parr G_2, \mathcal{G}'$.

$\multimap$-rule: The previous sequent must be $\mathcal{P}, D' \vdash G, \mathcal{G}'$, where $\mathcal{G} = D' \multimap G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D, D' \vdash_R G, \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R D' \multimap G, \mathcal{G}'$.

$\bigwedge$-rule: The previous sequent must be $\mathcal{P} \vdash G[y/x], \mathcal{G}'$, where $\mathcal{G} = \bigwedge x \,.\, G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D \vdash_R G[y/x], \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R \bigwedge x \,.\, G, \mathcal{G}'$.

$\bigvee$-rule: The previous sequent must be $\mathcal{P} \vdash G[t/x], \mathcal{G}'$, where $\mathcal{G} = \bigvee x \,.\, G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D \vdash_R G[t/x], \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R \bigvee x \,.\, G, \mathcal{G}'$.

This completes the proof. □

We may think of the above lemma as showing that we may add a formula $!D$ to an antecedent, a multiset of mixed clauses, $\mathcal{P}$ and retain the consequences of the original antecedent. Below we show that contraction is admissible, *i.e.*, adding $!D$ once will suffice, as adding it twice or more does not allow any new results to be derived.

LEMMA 6.12 (CONTRACTION) *Let $\mathcal{P}$ be a multiset of mixed clauses, $!D$ be a mixed clause and let $\mathcal{G}$ be a multiset of goal formulae. If $\mathcal{P}, !D, !D \vdash_R \mathcal{G}$, then $\mathcal{P}, !D \vdash_R \mathcal{G}$.*

PROOF
We proceed by induction on the height of the proof. The base case occurs when $\mathcal{G}$ is just a positive literal L, and it is clear that the lemma holds in this case. Hence we assume that the lemma is true for all sequents whose proof is no more than a given height.

Consider the step used to derive the current sequent. There are ten cases.

resolution: The previous sequent must be $\mathcal{P}'[\vec{t}/\vec{x}], !D, !D \vdash G, \mathcal{G}'$ where $\mathcal{P} = \mathcal{P}' \cup \{G' \multimap L'\}$, $\mathcal{G} = G, \mathcal{G}'$ and $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$. By the hypothesis we have $\mathcal{P}'[\vec{t}/\vec{x}], !D \vdash G, \mathcal{G}'$, and so we may apply the resolution rule to obtain $\mathcal{P}, !D \vdash_R \mathcal{G}$.

!-resolution: The previous sequent must be $\mathcal{P}[\vec{t}/\vec{x}]^g \cup [\vec{C}[\vec{t}/\vec{x}]] \cup \{!((G' \multimap L') \otimes \vec{C})[\vec{t}/\vec{x}]^g\}, !D, !D \vdash G, \mathcal{G}'$ where $\mathcal{P} = \mathcal{P}' \cup \{!((G' \multimap L') \otimes C)\}$, $\mathcal{G} = G, \mathcal{G}'$ and $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$. By the hypothesis we have $\mathcal{P}[\vec{t}/\vec{x}]^g \cup [\vec{C}[\vec{t}/\vec{x}]] \cup \{(!(G' \multimap L') \otimes \vec{C})[\vec{t}/\vec{x}]^g\}, !D \vdash G, \mathcal{G}'$ so we may apply the !-resolution rule to obtain $\mathcal{P}, !D \vdash_R \mathcal{G}$.

$\perp$-rule: The previous sequent must be $\mathcal{P}, !D, !D, L \vdash \mathcal{G}'$, where $\mathcal{G} = L^{\perp}, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D, L \vdash_R \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R L^{\perp}, \mathcal{G}'$.

& -rule: The previous sequents must be $\mathcal{P}', !D', !D' \vdash G'_1, \mathcal{G}'_1$ and $\mathcal{P}'', !D'', !D'' \vdash G'_2, \mathcal{G}'_2$, where $\mathcal{G} = G_1 \mathbin{\&} G_2, \mathcal{G}_1, \mathcal{G}_2$, $\mathcal{P}'$, $\mathcal{P}''$ are distinct copies of $\mathcal{P}$, $!D'$ and $!D''$ are distinct copies of $!D$ and G'_1, $\mathcal{G}'_1$, G'_2, $\mathcal{G}'_2$ are the correspondingly updated versions of G_1, $\mathcal{G}_1$, G_2, $\mathcal{G}_2$ respectively, and so by the hypothesis we have $\mathcal{P}, !D' \vdash_R G'_1, \mathcal{G}'_1$ and $\mathcal{P}, !D'' \vdash_R G'_2, \mathcal{G}'_2$, and so $\mathcal{P}, !D \vdash_R G_1 \mathbin{\&} G_2, \mathcal{G}_1, \mathcal{G}_2$.

$\otimes$-rule: The previous sequents must be $\mathcal{P}^I, \mathcal{P}_1 \vdash G_1, \mathcal{G}_1$ and $\mathcal{P}^I, \mathcal{P}_2 \vdash G_2, \mathcal{G}_2$, where $\mathcal{P}_1$, $\mathcal{P}_2$ is an expansion of $\mathcal{P} \cup \{!D, !D\}$ and $\mathcal{G} = G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$, and as $\mathcal{P}_1$, $\mathcal{P}_2$ is clearly also an expansion of $\mathcal{P} \cup \{!D\}$, we have $\mathcal{P}, !D \vdash_R G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$.

$\oplus$-rule: The previous sequent must be either $\mathcal{P}, !D, !D \vdash G_1, \mathcal{G}'$ or $\mathcal{P}, !D, !D \vdash G_2, \mathcal{G}'$, where $\mathcal{G} = G_1 \oplus G_2, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}_1, !D \vdash_R G_1, \mathcal{G}'$ or $\mathcal{P}, !D \vdash_R G_2, \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R G_1 \oplus G_2, \mathcal{G}'$.

$\maltese$-rule: The previous sequent must be $\mathcal{P}, !D, !D \vdash G_1, G_2, \mathcal{G}'$, where $\mathcal{G} = G_1 \maltese G_2, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D \vdash_R G_1, G_2, \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R G_1 \maltese G_2, \mathcal{G}'$.

$\multimap$-rule: The previous sequent must be $\mathcal{P}, !D, !D, D' \vdash G, \mathcal{G}'$, where $\mathcal{G} = D' \multimap G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D, D' \vdash_R G, \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R D' \multimap G, \mathcal{G}'$.

$\bigwedge$-rule: The previous sequent must be $\mathcal{P}, !D, !D \vdash G[y/x], \mathcal{G}'$, where $\mathcal{G} = \bigwedge x \,.\, G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D \vdash_R G[y/x], \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R \bigwedge x \,.\, G, \mathcal{G}'$.

$\bigvee$-rule: The previous sequent must be $\mathcal{P}, !D, !D \vdash G[t/x], \mathcal{G}'$, where $\mathcal{G} = \bigvee x \,.\, G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D \vdash_R G[t/x], \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R \bigvee x \,.\, G, \mathcal{G}'$.

This completes the proof. □

Next we show that we may strengthen the antecedent itself. Indeed, the following lemma is analogous to the !-L rule of linear sequent calculus.

LEMMA 6.13 (!-LEFT) *Let $\mathcal{P}$ be a multiset of mixed clauses, and let $\mathcal{G}$ be a multiset of goal formulae. Let D be a linear clause. If $\mathcal{P} \cup \{D\}^n \vdash_R \mathcal{G}$ for some integer $n \geq 0$ then $\mathcal{P}, !D \vdash_R \mathcal{G}$.*

PROOF
The $n = 0$ case was shown in Lemma 6.11. Consider the case for $n = 1$.

We proceed by induction on the height of the proof. The base case occurs when $\mathcal{G}$ is a single positive literal L, and it is clear that the lemma holds in this case.

Hence the induction hypothesis is that the lemma holds for all sequents whose resolution proof is no more than a given height. Consider the step used to derive the current sequent. There are ten cases.

resolution: There are two cases here. If the clause used is in $\mathcal{P}$, then the previous sequent is $\mathcal{P}'[\vec{t}/\vec{x}], D \vdash G, \mathcal{G}'$ where $\mathcal{P} = \mathcal{P}' \cup \{G' \multimap L'\}$, $\mathcal{G} = G, \mathcal{G}'$ and $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$.

By the hypothesis we have $\mathcal{P}'[\vec{t}/\vec{x}], !D \vdash G, \mathcal{G}'$, and so we may apply the resolution rule to obtain $\mathcal{P}, !D \vdash_R \mathcal{G}$. Otherwise $\mathcal{P} = \mathcal{P}'$, *i.e.*, D is the clause used, and so as above, $\mathcal{P}[\vec{t}/\vec{x}] \vdash_R G, \mathcal{G}'$, and by Lemma 6.11 we have $\mathcal{P}[\vec{t}/\vec{x}], !(G' \multimap L')[\vec{t}/\vec{x}] \vdash_R G, \mathcal{G}'$, to which we may apply the !-resolution rule, and so $\mathcal{P}, !D \vdash_R L, \mathcal{G}'$.

!-resolution: As D is not of the form $!F$, D cannot be the clause used, and so the the previous sequent is $\mathcal{P}[\vec{t}/\vec{x}]^g \cup [\vec{C}[\vec{t}/\vec{x}]] \cup \{!((G' \multimap L') \otimes \vec{C})[\vec{t}/\vec{x}]^g\}, D \vdash G, \mathcal{G}'$ where $\mathcal{P} = \mathcal{P}' \cup \{!((G' \multimap L') \otimes C)\}$, $\mathcal{G} = G, \mathcal{G}'$ and $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$. By the hypothesis we have $\mathcal{P}[\vec{t}/\vec{x}]^g \cup [\vec{C}[\vec{t}/\vec{x}]] \cup \{(!(G' \multimap L') \otimes \vec{C})[\vec{t}/\vec{x}]^g\}, !D \vdash G, \mathcal{G}'$ and so we may apply the !-resolution rule to obtain $\mathcal{P}, !D \vdash_R L, \mathcal{G}'$.

$\perp$-rule: The previous sequent must be $\mathcal{P}, D, L \vdash \mathcal{G}'$, where $\mathcal{G} = L^{\perp}, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D, L \vdash_R \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R L^{\perp}, \mathcal{G}'$.

& -rule: The previous sequents must be $\mathcal{P}', D' \vdash G_1', \mathcal{G}_1'$ and $\mathcal{P}'', D'' \vdash G_2', \mathcal{G}_2'$, where $\mathcal{G} = G_1 \,\&\, G_2, \mathcal{G}_1, \mathcal{G}_2$, $\mathcal{P}'$, $\mathcal{P}''$ are distinct copies of $\mathcal{P}$, D' and D'' are distinct copies of D and G_1', $\mathcal{G}_1'$, G_2', $\mathcal{G}_2'$ are the correspondingly updated versions of G_1, $\mathcal{G}_1$, G_2, $\mathcal{G}_2$ respectively, and so by the hypothesis we have $\mathcal{P}, !D' \vdash_R G_1', \mathcal{G}_1'$ and $\mathcal{P}, !D'' \vdash_R G_2', \mathcal{G}_2'$, and so $\mathcal{P}, !D \vdash_R G_1 \,\&\, G_2, \mathcal{G}_1, \mathcal{G}_2$.

$\otimes$-rule: Without loss of generality we may assume that the previous sequents are $\mathcal{P}^I, \mathcal{P}_1, D \vdash G_1, \mathcal{G}_1$ and $\mathcal{P}^I, \mathcal{P}_2 \vdash G_2, \mathcal{G}_2$, where $\mathcal{P}_1$, $\mathcal{P}_2$ is an expansion of $\mathcal{P} \cup \{D\}$ and $\mathcal{G} = G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$, and so by the hypothesis we have $\mathcal{P}^I, \mathcal{P}_1, !D \vdash_R G_1, \mathcal{G}_1$ and $\mathcal{P}^I, \mathcal{P}_2 \vdash_R G_2, \mathcal{G}_2$, and by Lemma 6.11 this implies that $\mathcal{P}^I, \mathcal{P}_2, !D \vdash_R G_2, \mathcal{G}_2$, and so $\mathcal{P}, !D \vdash_R G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$.

$\oplus$-rule: The previous sequent must be either $\mathcal{P}, D \vdash G_1, \mathcal{G}'$ or $\mathcal{P}, D \vdash G_2, \mathcal{G}'$, where $\mathcal{G} = G_1 \oplus G_2, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}_1, !D \vdash_R G_1, \mathcal{G}'$ or $\mathcal{P}, !D \vdash_R G_2, \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R G_1 \oplus G_2, \mathcal{G}'$.

$\parr$-rule: The previous sequent must be $\mathcal{P}, D \vdash G_1, G_2, \mathcal{G}'$, where $\mathcal{G} = G_1 \parr G_2, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D \vdash_R G_1, G_2, \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R G_1 \parr G_2, \mathcal{G}'$.

$\multimap$-rule: The previous sequent must be $\mathcal{P}, D, D' \vdash G, \mathcal{G}'$, where $\mathcal{G} = D' \multimap G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D, D' \vdash_R G, \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R D' \multimap G, \mathcal{G}'$.

$\bigwedge$-rule: The previous sequent must be $\mathcal{P}, D \vdash G[y/x], \mathcal{G}'$, where $\mathcal{G} = \bigwedge x\,.\,G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D \vdash_R G[y/x], \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R \bigwedge x\,.\,G, \mathcal{G}'$.

$\bigvee$-rule: The previous sequent must be $\mathcal{P}, D \vdash G[t/x], \mathcal{G}'$, where $\mathcal{G} = \bigvee x\,.\,G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, !D \vdash_R G[t/x], \mathcal{G}'$, and so $\mathcal{P}, !D \vdash_R \bigvee x\,.\,G, \mathcal{G}'$.

Thus the lemma holds for $n = 1$. A simple inductive argument then establishes the general case. This completes the proof. □

One interpretation of this result is that we may replace an occurrence of the resolution rule using the clause D with an occurrence of the !-resolution rule using the clause $!D$. Note that these three lemmas imply that the set-theoretic properties of formulae commencing with ! in arbitrary linear proofs are preserved in extended resolution proofs. It is this property which allows us to perform "mixed" intuitionistic and linear deduction in this system. This justifies our remark above that a formula with a ! at the front behaves in an antecedent precisely the same way as the same formula in an antecedent in the intuitionistic sequent calculus.

Finally we show a lemma of some technical importance; essentially this is that an occurrence of the resolution rule may be replaced by the !-resolution rule on a corresponding program. This correspondence is given by the mapping $-^!$, defined below.

DEFINITION 6.14 (THE MAPPING $-^!$) *Let $\mathcal{D}$ be a multiset of mixed clauses. We define a mapping $-^!$ from multisets of mixed clauses to multisets of mixed clauses as follows:*

$$\mathcal{D}^! = \{!\bigotimes_{C\in[\mathcal{D}]^L} C\} \cup \mathcal{D}^I \qquad \Box$$

Note the similarity between this definition and the definition of $[!D]$. We may think of the mapping $-^!$ as converting a set of linear clauses into a single mixed clause, and hence replacing uses of the resolution rule with the !-resolution rule. Below we show that this mapping preserves provability via resolution proofs.

LEMMA 6.15 (SOUNDNESS OF $-^!$) *Let $\mathcal{P}$ be a multiset of mixed clauses, and let $\mathcal{G}$ be a multiset of goal formulae. Let $\mathcal{D}$ be a multiset of mixed clauses. If $\mathcal{P}, \mathcal{D} \vdash_R \mathcal{G}$ then $\mathcal{P}, \mathcal{D}^! \vdash_R \mathcal{G}$.*

PROOF
We proceed by induction on the height of the proof. The base case occurs when $\mathcal{G}$ is a single positive literal L, and it is clear that the lemma holds in this case.

Hence the induction hypothesis is that the lemma holds for all sequents whose resolution proof is no more than a given height. Consider the step used to derive the current sequent. There are ten cases.

resolution: There are two cases here. If the clause used is in $\mathcal{P}$, then the previous sequent is $\mathcal{P}'[\vec{t}/\vec{x}], \mathcal{D}[\vec{t}/\vec{x}] \vdash G, \mathcal{G}'$ where $\mathcal{P} = \mathcal{P}' \cup \{G' \multimap L'\}$, $\mathcal{G} = G, \mathcal{G}'$ and $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$. By the hypothesis we have $\mathcal{P}'[\vec{t}/\vec{x}], \mathcal{D}^![\vec{t}/\vec{x}] \vdash G, \mathcal{G}'$, and so we may apply the resolution rule to obtain $\mathcal{P}, \mathcal{D}^! \vdash_R \mathcal{G}$. Otherwise $\mathcal{P} = \mathcal{P}'$, *i.e.*, the clause used is in $\mathcal{D}$, and so as above, $\mathcal{P}[\vec{t}/\vec{x}], \mathcal{D}'[\vec{t}/\vec{x}] \vdash_R G, \mathcal{G}'$ where $\mathcal{D} = \mathcal{D}' \cup \{G' \multimap L'\}$, and by Lemma 6.11 we have $\mathcal{P}[\vec{t}/\vec{x}], \mathcal{D}'[\vec{t}/\vec{x}], \mathcal{D}^![\vec{t}/\vec{x}] \vdash_R G, \mathcal{G}'$, to which we may apply the !-resolution rule to get $\mathcal{P}, \mathcal{D}^! \vdash_R L, \mathcal{G}'$.

!-resolution: As the clause used must be either an element of $\mathcal{P}$ or an element of $\mathcal{D}^I$, let $\mathcal{P}' \cup \{!((G' \multimap L') \otimes \vec{C})\} = \mathcal{P} \cup \mathcal{D}^I$. Hence the previous sequent is $\mathcal{P}'[\vec{t}/\vec{x}]^g, [\vec{C}[\vec{t}/\vec{x}]] \cup$

$\{!((G' \multimap L')\otimes\vec{C})[\vec{t}/\vec{x}]^g\}, \mathcal{D}^L[\vec{t}/\vec{x}]^g \vdash G, \mathcal{G}'$ where $\mathcal{G} = G, \mathcal{G}'$ and $G \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$. By the hypothesis we have $\mathcal{P}'[\vec{t}/\vec{x}]^g, [\vec{C}[\vec{t}/\vec{x}]] \cup \{!((G' \multimap L')\otimes\vec{C})[\vec{t}/\vec{x}]^g\}, (\mathcal{D}^L)^![\vec{t}/\vec{x}]^g \vdash G, \mathcal{G}'$ and so we may apply the !-resolution rule to obtain $\mathcal{P}, \mathcal{D}^! \vdash_R L, \mathcal{G}'$.

& -rule: The previous sequents must be $\mathcal{P}', \mathcal{D}' \vdash G_1', \mathcal{G}_1'$ and $\mathcal{P}'', \mathcal{D}'' \vdash G_2', \mathcal{G}_2'$, where $\mathcal{G} = G_1 \,\&\, G_2, \mathcal{G}_1, \mathcal{G}_2$, $\mathcal{P}'$, $\mathcal{P}''$ are distinct copies of $\mathcal{P}$, $\mathcal{D}'$ and $\mathcal{D}''$ are distinct copies of $\mathcal{D}$ and G_1', $\mathcal{G}_1'$, G_2', $\mathcal{G}_2'$ are the correspondingly updated versions of G_1, $\mathcal{G}_1$, G_2, $\mathcal{G}_2$ respectively, and so by the hypothesis we have $\mathcal{P}, \mathcal{D}'^! \vdash_R G_1', \mathcal{G}_1'$ and $\mathcal{P}, \mathcal{D}''^! \vdash_R G_2', \mathcal{G}_2'$, and so $\mathcal{P}, \mathcal{D}^! \vdash_R G_1 \,\&\, G_2, \mathcal{G}_1, \mathcal{G}_2$.

$\perp$-rule: The previous sequent must be $\mathcal{P}, \mathcal{D}, L \vdash \mathcal{G}'$, where $\mathcal{G} = L^\perp, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, \mathcal{D}^!, L \vdash_R \mathcal{G}'$, and so $\mathcal{P}, \mathcal{D}^! \vdash_R L^\perp, \mathcal{G}'$.

$\otimes$-rule: The previous sequents must be $\mathcal{P}^I, \mathcal{P}_1, \mathcal{D}^I \vdash G_1, \mathcal{G}_1$ and $\mathcal{P}^I, \mathcal{P}_2, \mathcal{D}^I \vdash G_2, \mathcal{G}_2$, where $\mathcal{P}_1$, $\mathcal{P}_2$ is an expansion of $\mathcal{P} \cup \mathcal{D}$ and $\mathcal{G} = G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$, and as an expansion of $\mathcal{D}$ is an expansion of $\mathcal{D}^!$, we may conclude that $\mathcal{P}, \mathcal{D}^! \vdash_R G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$.

$\oplus$-rule: The previous sequent must be either $\mathcal{P}, \mathcal{D} \vdash G_1, \mathcal{G}'$ or $\mathcal{P}, \mathcal{D} \vdash G_2, \mathcal{G}'$, where $\mathcal{G} = G_1 \oplus G_2, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}_1, \mathcal{D}^! \vdash_R G_1, \mathcal{G}'$ or $\mathcal{P}, \mathcal{D}^! \vdash_R G_2, \mathcal{G}'$, and so $\mathcal{P}, \mathcal{D}^! \vdash_R G_1 \oplus G_2, \mathcal{G}'$.

$\invamp$-rule: The previous sequent must be $\mathcal{P}, \mathcal{D} \vdash G_1, G_2, \mathcal{G}'$, where $\mathcal{G} = G_1 \invamp G_2, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, \mathcal{D}^! \vdash_R G_1, G_2, \mathcal{G}'$, and so $\mathcal{P}, \mathcal{D}^! \vdash_R G_1 \invamp G_2, \mathcal{G}'$.

$\multimap$-rule: The previous sequent must be $\mathcal{P}, \mathcal{D}, D' \vdash G, \mathcal{G}'$, where $\mathcal{G} = D' \multimap G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, \mathcal{D}^!, D' \vdash_R G, \mathcal{G}'$, and so $\mathcal{P}, \mathcal{D}^! \vdash_R D' \multimap G, \mathcal{G}'$.

$\bigwedge$-rule: The previous sequent must be $\mathcal{P}, \mathcal{D} \vdash G[y/x], \mathcal{G}'$, where $\mathcal{G} = \bigwedge x . G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, \mathcal{D}^! \vdash_R G[y/x], \mathcal{G}'$, and so $\mathcal{P}, \mathcal{D}^! \vdash_R \bigwedge x . G, \mathcal{G}'$.

$\bigvee$-rule: The previous sequent must be $\mathcal{P}, \mathcal{D} \vdash G[t/x], \mathcal{G}'$, where $\mathcal{G} = \bigvee x . G, \mathcal{G}'$, and so by the hypothesis we have $\mathcal{P}, \mathcal{D}^! \vdash_R G[t/x], \mathcal{G}'$, and so $\mathcal{P}, \mathcal{D}^! \vdash_R \bigvee x . G, \mathcal{G}'$.

This completes the proof. □

Now we show that extended resolution proofs are sound with respect to linear deduction.

PROPOSITION 6.16 (SOUNDNESS OF EXTENDED RESOLUTION PROOFS) *Let $\mathcal{P}$ be a multiset of definite formulae, and let $\mathcal{G}$ be a multiset of goal formulae. If $[\mathcal{P}] \vdash_R \mathcal{G}$, then $\mathcal{P} \vdash \mathcal{G}$ has a simple locally LR proof.*

PROOF
By Proposition 6.3 it will suffice to show $\mathcal{P} \vdash \mathcal{G}$ is provable.

We proceed by induction on the size of the proof. The base case occurs when $\mathcal{G}$ is just a positive literal L such that either $[\mathcal{P}]^L = \{L'\}$ and $L = L'[\vec{t}/\vec{x}]$, or $[\mathcal{P}]^L = \emptyset$ and there is a clause $!L' \in [\mathcal{P}]^I$ such that $L = L'[\vec{t}/\vec{x}]$, and in either case it is clear that $\mathcal{P} \vdash L$ is provable.

Hence the induction hypothesis is that the proposition holds for all sequents whose extended resolution proofs no more than a given height. Consider the step used to derive the current sequent. There are several cases.

resolution: In this case we have that $[\mathcal{P}'][\vec{t}/\vec{x}] \vdash_R G', \mathcal{G}'$ where $[\mathcal{P}'] \cup \{G \multimap L\} = [\mathcal{P}]$, $G' \multimap L' = (G \multimap L)[\vec{t}/\vec{x}]$, and $\mathcal{G} = \mathcal{G}' \cup \{L'\}$. By the hypothesis, $\mathcal{P}'[\vec{t}/\vec{x}] \vdash G', \mathcal{G}'$ has a proof, and as $L' \vdash L'$ is an initial sequent, we may apply the $\multimap$-L rule to derive $\mathcal{P}'[\vec{t}/\vec{x}], G' \multimap L' \vdash \mathcal{G}', L'$. Then we may apply $\bigwedge$-L, & -L and/or $\otimes$-L a number of times to get a proof of $\mathcal{P} \vdash \mathcal{G}$.

!-resolution: In this case we have that $[\mathcal{P}'][\vec{t}/\vec{x}]^g, [\vec{C}[\vec{t}/\vec{x}]], !((G \multimap L) \otimes \vec{C})[\vec{t}/\vec{x}]^g \vdash_R G', \mathcal{G}'$ where $[\mathcal{P}'] \cup \{!((G \multimap L) \otimes \vec{C})\} = [\mathcal{P}]$, $G' \multimap L' = (G \multimap L)[\vec{t}/\vec{x}]$, and $\mathcal{G} = \mathcal{G}' \cup \{L'\}$. By the hypothesis, $\mathcal{P}'[\vec{t}/\vec{x}]^g, \vec{C}[\vec{t}/\vec{x}], !D[\vec{t}/\vec{x}]^g \vec{C})[\vec{t}/\vec{x}]^g \vdash G', \mathcal{G}'$ is provable, where $[!D] = !((G \multimap L) \otimes \vec{C})$, and as $L' \vdash L'$ is an initial sequent, we may apply the $\multimap$-L rule to derive $\mathcal{P}'[\vec{t}/\vec{x}]^g, G' \multimap L', \vec{C}[\vec{t}/\vec{x}], !D[\vec{t}/\vec{x}]^g \vdash \mathcal{G}', L'$. Then we may apply $\bigwedge$-L, $\otimes$-L, !-L and/or & -L a number of times to get a proof of $\mathcal{P}'[\vec{t}/\vec{x}]^g, D[\vec{t}/\vec{x}]^g, !D[\vec{t}/\vec{x}]^g \vdash \mathcal{G}', L'$, and we then apply !-L and contraction to get a proof of $\mathcal{P}'[\vec{t}/\vec{x}]^g, !D[\vec{t}/\vec{x}]^g \vdash \mathcal{G}', L'$. We may then apply the $\bigwedge$-L, $\otimes$-L, & -L and/or !-L a number of times to get a proof of $\mathcal{P} \vdash \mathcal{G}$.

$\perp$-rule: In this case we have that $[\mathcal{P}], L \vdash_R \mathcal{G}'$ where $\mathcal{G} = \{L^\perp\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P}, L \vdash \mathcal{G}'$ is provable, and so $\mathcal{P} \vdash L^\perp, \mathcal{G}'$ is provable.

& -rule: In this case we have that $\mathcal{P}_1 \vdash_R G_1', \mathcal{G}''$ and $\mathcal{P}_2 \vdash_R G_2', \mathcal{G}'''$ where $\mathcal{G} = \{G_1 \,\&\, G_2\} \cup \mathcal{G}'$, $\mathcal{P}_1$, $\mathcal{P}_2$ are distinct copies of $\mathcal{P}$ and G_1', G_2', $\mathcal{G}''$ and $\mathcal{G}'''$ are the correspondingly updated versions of G_1, G_2 and $\mathcal{G}'$ respectively, and so by the hypothesis $\mathcal{P}_1' \vdash G_1', \mathcal{G}''$ and $\mathcal{P}_2' \vdash G_2', \mathcal{G}'''$ are provable, where $\mathcal{P}_1'$ and $\mathcal{P}_2'$ are distinct copies of $\mathcal{P}$. It follows that $\mathcal{P} \vdash G_1, \mathcal{G}'$ and $\mathcal{P} \vdash G_2, \mathcal{G}'$ are provable, and so and so $\mathcal{P} \vdash G_1 \,\&\, G_2, \mathcal{G}'$ is provable.

$\otimes$-rule: In this case we have that $[\mathcal{P}]^I, \mathcal{P}_1 \vdash_R G_1, \mathcal{G}_1$ and $[\mathcal{P}]^I, \mathcal{P}_2 \vdash_R G_2, \mathcal{G}_2$ where $\mathcal{P}_1, \mathcal{P}_2$ is an expansion of $[\mathcal{P}]$ and $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$, and so by the hypothesis $\mathcal{P}^I, \mathcal{P}_1 \vdash G_1, \mathcal{G}_1$ and $\mathcal{P}^I, \mathcal{P}_2 \vdash G_2, \mathcal{G}_2$ are provable, as $[\mathcal{P}_1] = \mathcal{P}_1$ and $[\mathcal{P}_2] = \mathcal{P}_2$, and so $\mathcal{P}^I, \mathcal{P}_1, \mathcal{P}_2 \vdash G_1 \otimes G_2, \mathcal{G}$ is provable. As in the !-resolution case, we may then apply the rules $\bigwedge$-L, !-L, & -L and/or $\otimes$-L a number of times to the latter sequent to obtain a proof of $\mathcal{P} \vdash G_1 \otimes G_2, \mathcal{G}$.

$\oplus$-rule: In this case we have that $[\mathcal{P}] \vdash_R G_1, \mathcal{G}'$ or $[\mathcal{P}] \vdash_R G_2, \mathcal{G}'$ where $\mathcal{G} = \{G_1 \oplus G_2\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P} \vdash G_1, \mathcal{G}'$ or $\mathcal{P} \vdash G_2, \mathcal{G}'$ is provable, and so $\mathcal{P} \vdash G_1 \maltese G_2, \mathcal{G}'$ is provable.

$\maltese$-rule: In this case we have that $[\mathcal{P}] \vdash_R G_1, G_2, \mathcal{G}'$ where $\mathcal{G} = \{G_1 \maltese G_2\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P} \vdash G_1, G_2, \mathcal{G}'$ is provable, and so $\mathcal{P} \vdash G_1 \,\&\, G_2, \mathcal{G}'$ is provable.

$\multimap$-rule: In this case we have that $[\mathcal{P}], [D]^! \vdash_R G, \mathcal{G}'$ where $\mathcal{G} = \{D \multimap G\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P}, D \vdash G, \mathcal{G}'$ is provable, and so $\mathcal{P} \vdash D \multimap G, \mathcal{G}'$ is provable.

$\bigwedge$-rule: In this case we have that $[\mathcal{P}] \vdash_R G[y/x], \mathcal{G}'$ where $\mathcal{G} = \{\bigwedge x.G\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P} \vdash G[y/x], \mathcal{G}'$ is provable, and so $\mathcal{P} \vdash \bigwedge x\,.\,G, \mathcal{G}'$ is provable.

$\bigvee$-rule: In this case we have that $[\mathcal{P}] \vdash_R G[t/x], \mathcal{G}'$ where $\mathcal{G} = \{\bigvee x\,.\,G\} \cup \mathcal{G}'$, and so by the hypothesis $\mathcal{P} \vdash G[y/x], \mathcal{G}'$ is provable, and so $\mathcal{P} \vdash \bigvee x\,.\,G, \mathcal{G}'$ is provable.

We remark that for this fragment of linear logic we have, by Theorem 3.18, that for any linear proof there is a corresponding locally LR proof. This completes the proof. □

Now we turn to the completeness of extended resolution proofs.

Proposition 6.17 (Completeness of Extended Resolution Proofs) *Let $\mathcal{P}$ be a multiset of definite formulae, and let $\mathcal{G}$ be a multiset of goal formulae. If $\mathcal{P} \vdash \mathcal{G}$ has a simple locally LR proof, then $[\mathcal{P}] \vdash_R \mathcal{G}$.*

Proof

We proceed by induction on the length of the proof of $\mathcal{P} \vdash \mathcal{G}$.

The base case occurs when $\mathcal{P} \vdash \mathcal{G}$ is initial, *i.e.* $\mathcal{P} = \mathcal{G} = \{L\}$ for some positive literal L, and clearly the proposition holds in this case.

Hence we assume that the proposition holds for all sequents whose proof is of no more than a given height. Consider the rule used to derive $\mathcal{P} \vdash \mathcal{G}$. By Lemma 6.2, we need only consider the rules $\otimes$-L, $\&$-L, $\bigwedge$-L, $\multimap$-L, !-L, C?-L, W?-L, $\otimes$-R, $\&$-R, $\oplus$-R, $⅋$-R, $\multimap$-R, $\bigwedge$-R and $\bigvee$-R.

$\otimes$-L: The previous sequent must be $D_1, D_2, \mathcal{P}' \vdash \mathcal{G}$ where $\{D_1 \otimes D_2\} \cup \mathcal{P}' = \mathcal{P}$, and so by the hypothesis, $[D_1], [D_2], [\mathcal{P}'] \vdash_R \mathcal{G}$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\&$-L: The previous sequent must be either $D_1, \mathcal{P}' \vdash \mathcal{G}$ or $D_2, \mathcal{P}' \vdash \mathcal{G}$ where $\{D_1 \& D_2\} \cup \mathcal{P}' = \mathcal{P}$, and so by the hypothesis, either $[D_1], [\mathcal{P}'] \vdash_R \mathcal{G}$ or $[D_2], [\mathcal{P}'] \vdash_R \mathcal{G}$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\bigwedge$-L: The previous sequent must be $D[t/x], \mathcal{P}' \vdash \mathcal{G}$, where $\{\bigwedge x\,.\,D\} \cup \mathcal{P}' = \mathcal{P}$, and so by the hypothesis we have that $[D[t/x], \mathcal{P}'] \vdash_R \mathcal{G}$, *i.e.,* $[D[t/x]] \cup [\mathcal{P}'] \vdash_R \mathcal{G}$. Hence by the instance property $[D] \cup [\mathcal{P}'] \vdash_R \mathcal{G}$ (the resolution proof of the induction hypothesis will suffice) and so $[\bigwedge x\,.\,D, \mathcal{P}'] \vdash_R \mathcal{G}$.

$\multimap$-L: The previous sequent must be $\mathcal{P}' \vdash G, \mathcal{G}'$ where $\{G \multimap L\} \cup \mathcal{P}' = \mathcal{P}$ and $\{L\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}'] \vdash_R G, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

!-L: The previous sequent must be $\mathcal{P}', D \vdash G, \mathcal{G}'$ where $\{!D\} \cup \mathcal{P}' = \mathcal{P}$, and so by the hypothesis, $[\mathcal{P}'], [D] \vdash_R G, \mathcal{G}'$, and by Lemma 6.15 $[\mathcal{P}'], [D]^! \vdash_R G, \mathcal{G}'$, *i.e.* $[\mathcal{P}]^! \vdash_R \mathcal{G}$.

C?-L: In this case the result follows by Lemma 6.12.

$W?$-L: In this case the result follows by Lemma 6.11.

$\otimes$-R: The previous sequents must be $\mathcal{P}_1 \vdash G_1, \mathcal{G}_1$ and $\mathcal{P}_2 \vdash G_2, \mathcal{G}_2$, where $\mathcal{P}_1 \cup \mathcal{P}_2 = \mathcal{P}$ and $\{G_1 \otimes G_2\}, \mathcal{G}_1 \cup \mathcal{G}_2 = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}_1] \vdash_R G_1, \mathcal{G}_1$ and $[\mathcal{P}_2] \vdash_R G_2, \mathcal{G}_2$. Now by Lemma 6.11 this implies that $[\mathcal{P}_1], [\mathcal{P}_2]^I \vdash_R G_1, \mathcal{G}_1$ and $[\mathcal{P}_2], [\mathcal{P}_1]^I \vdash_R G_2, \mathcal{G}_2$, and so we get that $[\mathcal{P}_1]^I, [\mathcal{P}_2]^I, [\mathcal{P}_1], [\mathcal{P}_2] \vdash_R G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$, which by Lemma 6.12 gives us $[\mathcal{P}_1], [\mathcal{P}_2] \vdash_R G_1 \otimes G_2, \mathcal{G}_1, \mathcal{G}_2$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\perp$-R: The previous sequent must be $\mathcal{P}, L \vdash \mathcal{G}'$, where $\{L^\perp\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}], L \vdash_R G, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

& -R: The previous sequents must be $\mathcal{P} \vdash G_1, \mathcal{G}'$ and $\mathcal{P} \vdash G_2, \mathcal{G}'$, where $\{G_1 \,\&\, G_2\} \cup \mathcal{G}' = \mathcal{G}$, and so the sequents $\mathcal{P}' \vdash G_1', \mathcal{G}''$ and $\mathcal{P}'' \vdash G_2', \mathcal{G}'''$ are provable, where $\mathcal{P}'$ and $\mathcal{P}''$ are distinct copies of $\mathcal{P}$ and G_1', G_2', $\mathcal{G}''$ and $\mathcal{G}'''$ are the correspondingly updated versions of G_1, G_2 and $\mathcal{G}'$ respectively, and so by the hypothesis, $[\mathcal{P}'] \vdash_R G_1', \mathcal{G}''$ and $[\mathcal{P}''] \vdash_R G_2', \mathcal{G}'''$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\oplus$-R: The previous sequents must be $\mathcal{P} \vdash G_1, \mathcal{G}'$ and $\mathcal{P} \vdash G_2, \mathcal{G}'$, where $\{G_1 \maltese G_2\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}] \vdash_R G_1, \mathcal{G}'$ or $[\mathcal{P}] \vdash_R G_2, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\maltese$-R: The previous sequent must be $\mathcal{P} \vdash G_1, G_2, \mathcal{G}'$, where $\{G_1 \maltese G_2\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}] \vdash_R G_1, G_2, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\multimap$-R: The previous sequent must be $\mathcal{P}, D \vdash G, \mathcal{G}'$, where $\{D \multimap G\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}], [D] \vdash_R G, \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\bigwedge$-R: The previous sequent must be $\mathcal{P} \vdash G[y/x], \mathcal{G}'$, where $\{\bigwedge x \,.\, G\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}] \vdash_R G[y/x], \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

$\bigvee$-R: The previous sequent must be $\mathcal{P} \vdash G[t/x], \mathcal{G}'$, where $\{\bigvee x \,.\, G\} \cup \mathcal{G}' = \mathcal{G}$, and so by the hypothesis, $[\mathcal{P}] \vdash_R G[t/x], \mathcal{G}'$, *i.e.* $[\mathcal{P}] \vdash_R \mathcal{G}$.

This completes the proof. □

Thus we find that extended resolution proofs are sound and complete with respect to linear provability.

THEOREM 6.18 *Let $\mathcal{P}$ be a multiset of extended definite formulae and let $\mathcal{G}$ be a set of extended goal formulae. Then $\mathcal{P} \vdash \mathcal{G}$ is provable iff $[\mathcal{P}] \vdash_R \mathcal{G}$.*

PROOF
The proof follows immediately from Propositions 6.3, 6.16 and 6.17. □

Note that this does not imply, for example, that $\vdash !(p \otimes !q) \multimap\!\circ\; !p \otimes !q$ (which is not provable in linear sequent calculus) is provable, even though $[!(p \otimes !q)] = \{\, !p, !q \,\} = [!p \otimes !q]$.

As noted above, we may think of the mixture of the resolution and !-resolution rules as providing mixed deduction. Also, we saw in Section 4 how purely linear deduction may be used. Hence it is natural to ask what purely intuitionistic deduction looks like in this setting. In particular, we show how intuitionistic proofs in the hereditary Harrop formulae fragment of first-order logic may be encoded as proofs in this fragment of linear logic.

In order so to do, we recall some definitions from [21], [20] for intuitionistic logic. We assume that we are dealing with first-order intuitionistic logic and first-order linear logic over some common language $\mathcal{L}$.

DEFINITION 6.19 (HEREDITARY HARROP FORMULAE) *Let A range over atomic formulae. We define the classes of* hereditary Harrop definite formulae *and* hereditary Harrop goal formulae *as follows:*

$$\begin{array}{lll} \textit{Definite formulae} & D & ::= & A \mid D \wedge D \mid G \supset A \mid \forall x \,.\, D \\[1em] \textit{Goal formulae} & G & ::= & A \mid G \vee G \mid G \wedge G \mid D \supset G \\ & & & \mid \forall x \,.\, G \mid \exists x \,.\, G \end{array}$$

A hereditary Harrop program *is a set of closed Hereditary Harrop definite formulae. A* hereditary Harrop goal *is a closed Hereditary Harrop goal formula.* □

Now we recall the definition of the relation $\vdash_o$ [21], [20]. In order to avoid notational confusion, we will use $|| - ||$ to denote the $[-]$ operation of [20].

DEFINITION 6.20 (HEREDITARY HARROP CLAUSAL DECOMPOSITION) *Let $\cup$ denote set union. We define a mapping $|| - ||$ from sets of hereditary Harrop definite formulae to sets of hereditary Harrop definite formulae inductively as follows:*

$$\begin{array}{lll} ||\mathcal{P}|| & =_{\text{def}} & \bigcup_{D \in \mathcal{P}} ||D|| \\ ||A|| & =_{\text{def}} & \{A\} \\ ||D_1 \wedge D_2|| & =_{\text{def}} & ||D_1|| \cup ||D_2|| \\ ||G \supset A|| & =_{\text{def}} & \{G \supset A\} \\ ||\forall x \,.\, D|| & =_{\text{def}} & \bigcup_{t \in \mathcal{U}'} ||D[t/x]|| \end{array}$$

where $\mathcal{U}'$ denotes the universe of ground terms. □

DEFINITION 6.21 (HEREDITARY HARROP UNIFORM PROOFS) *Let $\mathcal{P}$ be a hereditary Harrop program and let $\mathcal{G}$ be a hereditary Harrop goal. The relation $\vdash_o$ is the least relation satisfying:*

- *$\mathcal{P} \vdash_o A$ iff either $A \in ||\mathcal{P}||$ or there is some $G \supset A \in ||\mathcal{P}||$ such that $\mathcal{P} \vdash_o G$*
- *$\mathcal{P} \vdash_o G_1 \vee G_2$ iff $\mathcal{P} \vdash_o G_1$ or $\mathcal{P} \vdash_o G_2$*
- *$\mathcal{P} \vdash_o G_1 \wedge G_2$ iff $\mathcal{P} \vdash_o G_1$ and $\mathcal{P} \vdash_o G_2$*

- $\mathcal{P} \vdash_o D \supset G$ *iff* $\mathcal{P}, D \vdash_o G$
- $\mathcal{P} \vdash_o \exists x . G$ *iff* $\mathcal{P} \vdash_o G[t/x]$ *for some* $t \in \mathcal{U}'$
- $\mathcal{P} \vdash_o \forall x . G$ *iff* $\mathcal{P} \vdash_o G[y/x]$ *where* y *is not free in* $\mathcal{P}$

□

The following result was shown in [21].

THEOREM 6.22 (INTUITIONISTIC OPERATIONAL PROVABILITY) *Let $\mathcal{P}$ be a hereditary Harrop formula program and let G be a hereditary Harrop goal. Let $\vdash_I$ denote intuitionsitic provability. Then $\mathcal{P} \vdash_o G$ iff $\mathcal{P} \vdash_I G$.* □

Next we present the encoding of intuitionistic formulae in linear formulae, via two mappings, $(-)^+$ and $(-)^-$.

DEFINITION 6.23 (ENCODING) *Let D range over hereditary Harrop definite formulae, let $\mathcal{D}$ range over multisets of (linear) definite formulae and let G range over hereditary Harrop goals. We define the linear encoding $(D)^-$ of D and the linear encoding $(G)^+$ of G as follows:*

$$\begin{array}{lll}
(A)^+ & =_{\text{def}} & A \\
(G_1 \wedge G_2)^+ & =_{\text{def}} & (G_1)^+ \mathbin{\&} (G_2)^+ \\
(G_1 \vee G_2)^+ & =_{\text{def}} & (G_1)^+ \oplus (G_2)^+ \\
(\exists \vec{x} . G)^+ & =_{\text{def}} & \bigvee \vec{x} . (G)^+ \\
(\forall \vec{x} . G)^+ & =_{\text{def}} & \bigwedge \vec{x} . (G)^+ \\
(D \supset G)^+ & =_{\text{def}} & (\bigotimes (D)^-) \multimap (G)^+
\end{array}
\qquad
\begin{array}{lll}
\mathcal{P}^- & =_{\text{def}} & \bigcup_{D \in \mathcal{P}} (D)^- \\
(D)^- & =_{\text{def}} & \bigcup_{C \in \|D\|} \{(C)^-\} \\
(\forall \vec{x} . A)^- & =_{\text{def}} & ! \bigwedge \vec{x} . A \\
(\forall \vec{x} . G \supset A)^- & =_{\text{def}} & ! \bigwedge \vec{x} . (G)^+ \multimap L
\end{array}$$

where $\bigotimes \mathcal{D}$ denotes $\bigotimes_{C \in \mathcal{D}} C$. □

We may think of these encodings as operating at the level of proofs, rather than at the level of formulae. This is due to the way that an intuitionistic conjunction may be translated as either of the two linear conjunctions, depending on whether it occurs in a positive position or not. We now show that this encoding behaves as expected.

PROPOSITION 6.24 (RESOLUTION PROVABILITY UNDER ENCODING) *Let $\mathcal{P}$ be a hereditary Harrop program and let G be a hereditary Harrop goal formula. Then $\mathcal{P} \vdash_o G$ iff $(\mathcal{P})^- \vdash_R (G)^+$.*

Proof
We proceed by induction on the size of the proofs. The base case is when G is an atom, and it is clear that the proposition holds in this case.

The induction hypothesis is that the proposition holds for all proofs of no more than a given size. There are six cases.

A: $\mathcal{P} \vdash_o A$ iff either $A \in \|\mathcal{P}\|$, in which case we are done (this is the base of the induction), or there is some $G \supset A \in \|\mathcal{P}\|$ such that $\mathcal{P} \vdash_o G$, and by the hypothesis this is equivalent to $[(\mathcal{P})^-] \vdash_R (G)^+$. This in turn is equivalent to the existence of a clause $!(G' \multimap L') \in [\mathcal{P}]$ such that $(G)^+ \multimap L = (G' \multimap L')[\vec{t}/\vec{x}]$, for some $\vec{t}$, and $[(\mathcal{P})^-] \vdash_R (G)^+$, which is just $[(\mathcal{P})^-] \vdash_R (A)^+$.

$G_1 \vee G_2$: $\mathcal{P} \vdash_o G_1 \vee G_2$ iff $\mathcal{P} \vdash_o G_1$ or $\mathcal{P} \vdash_o G_2$ and by the hypothesis this is equivalent to $[(\mathcal{P})^-] \vdash_R (G_1)^+$ or $[(\mathcal{P})^-] \vdash_R (G_2)^+$, which in turn is equivalent to $[(\mathcal{P})^-] \vdash_R (G_1)^+ \oplus (G_2)^+$, which is just $[(\mathcal{P})^-] \vdash_R (G_1 \vee G_2)^+$.

$G_1 \wedge G_2$: $\mathcal{P} \vdash_o G_1 \wedge G_2$ iff $\mathcal{P} \vdash_o G_1$ and $\mathcal{P} \vdash_o G_2$ and by the hypothesis this is equivalent to $[(\mathcal{P})^-] \vdash_R (G_1)^+$ and $[(\mathcal{P})^-] \vdash_R (G_2)^+$, which in turn is equivalent to $[(\mathcal{P})^-] \vdash_R (G_1)^+ \mathbin{\&} (G_2)^+$ (by some renaming), which is just $[(\mathcal{P})^-] \vdash_R (G_1 \wedge G_2)^+$.

$\exists x\,.\,G$: $\mathcal{P} \vdash_o \exists x\,.\,G$ iff $\mathcal{P} \vdash_o G[t/x]$ for some $t \in \mathcal{U}'$, and by the hypothesis this is equivalent to $[(\mathcal{P})^-] \vdash_R (G[t/x])^+$, which in turn is equivalent to $[(\mathcal{P})^-] \vdash_R \bigvee x\,.\,(G)^+$, which is just $[(\mathcal{P})^-] \vdash_R (\exists x\,.\,G)^+$.

$\forall x\,.\,G$: $\mathcal{P} \vdash_o \forall x\,.\,G$ iff $\mathcal{P} \vdash_o G[y/x]$ where y does not occur free in P, and by the hypothesis this is equivalent to $[(\mathcal{P})^-] \vdash_R (G[y/x])^+$, which in turn is equivalent to $[(\mathcal{P})^-] \vdash_R \bigwedge x\,.\,(G)^+$, which is just $[(\mathcal{P})^-] \vdash_R (\forall x\,.\,G)^+$.

$D \supset G$: $\mathcal{P} \vdash_o D \supset G$ iff $\mathcal{P}, D \vdash_o G$, and by the hypothesis this is equivalent to $[(\mathcal{P} \cup \{D\})^-] \vdash_R (G)^+$, which in turn is equivalent to $[(\mathcal{P})^-] \vdash_R (\otimes(D)^-) \multimap (G)^+$, which is just $[(\mathcal{P})^-] \vdash_R (D \supset G)^+$.

This completes the proof. □

Thus we arrive at the following theorem.

Theorem 6.25 (Provabilty Under Encoding) *Let $\mathcal{P}$ be a hereditary Harrop program and G be a hereditary Harrop goal. $\mathcal{P} \vdash G$ is provable in intuitionistic logic iff $[(\mathcal{P})^-] \vdash (G)^+$ is provable in linear logic.*

Proof
By Theorem 6.23, $\mathcal{P} \vdash G$ is provable in intuitionistic logic iff $\mathcal{P} \vdash_o G$, and by Proposition 6.24 this is equivalent to $[(\mathcal{P})^-] \vdash_R (G)^+$, and by Theorem 6.18, this is equivalent to the provability of $(\mathcal{P})^- \vdash (G)^+$ in linear sequent calculus. □

7 Implementation Issues

In this section we sketch the design of an interpreter for linear logic programs. Essentially, the interpreter will execute resolution proofs, however as we noted in Sections 4 and 6, the definition of resolution proof does not provide a mechanism for calculating the appropriate splitting of the antecedent and succedent in the $\otimes$-rule. This problem is particularly acute in the presence of clauses whose top-level connective is !, and so we shall begin with this case here. Furthermore, for the purposes of this section we restrict our attention to *single-conclusioned* sequents.[21]

We proceed by presenting several examples, each of which will illustrate certain aspects of the definition of an interpreter. Full details of such an interpreter are provided in a forthcoming paper by the authors which describes a prototype implementation of logic programming for this fragment of linear logic.

EXAMPLE 1 First we consider only the $\otimes$- and $\&$ -rules. Suppose that we are faced with the endsequent

$$!(p \otimes q) \vdash (p \otimes q) \mathbin{\&} (p \otimes (q \otimes q)),$$

which is not provable in linear sequent calculus. We must attempt a resolution proof of

$$[!(p \otimes q)](= \{\, !(p \otimes q) \,\}) \vdash_R (p \otimes q) \mathbin{\&} (p \otimes (q \otimes q)),$$

and *fail.* According to the definition of resolution proof, we must identify a suitable expansion of $!(p \otimes q)$ and proceed to identify suitable splittings of that expansion: computationally, this is unacceptable. Our solution is to adopt a *lazy* approach, and to this end we permit the interpreter to construct the following *proto-proof* by modifying the rules of resolution proof:

$$\dfrac{\dfrac{\{\,!(p \otimes q)\,\} \vdash_R p \quad \{\,!(p \otimes q)\,\} \vdash_R q}{\{\,!(p \otimes q)\,\} \vdash_R p \otimes q}\otimes \qquad \dfrac{\{\,!(p \otimes q)\,\} \vdash^*_R p \quad \dfrac{\{!(p \otimes q)\} \vdash^*_R q \quad \{!(p \otimes q)\} \vdash^*_R q}{\{\,!(p \otimes q)\,\} \vdash^*_R q \otimes q}\otimes}{\{\,!(p \otimes q)\,\} \vdash^*_R p \otimes (q \otimes q)}\otimes}{\{\,!(p \otimes q)\,\} \vdash^*_R (p \otimes q) \mathbin{\&} (p \otimes (q \otimes q))}\&$$

Define the construction of a *path* in such a proto-proof as follows:

- The ensequent is in every path;
- Traverse the tree towards the leaves, starting at the ensequent:
  - Whenever a &-rule is reached, choose a branch and proceed;
  - Whenever a $\otimes$-rule is reached, proceed along both branches;
- Continue until all branches of the path have reached a leaf.

[21]This forces us to abandon $\maltese$.

The proto-proof determines a resolution proof just in case for each possible such path in it, there is an expansion of the antecedent that is compatible with the leaves in the path. For example, the proto-proof given above has a path marked by asterisks (*). For this path, there are 2 occurrences of q in the leaves, so that we need the expansion $\{p, p, q, q\}$ of $\{!(p \otimes q)\}$. However, there is only 1 occurrence of p in the leaves of this path, so this expansion is not suitable. We conclude that this proto-proof does not determine a resolution proof of the given endsequent.

EXAMPLE 2 Now we introduce the $\multimap$-rule. Suppose that we are faced with the endsequent

$$!(p \otimes q) \vdash (q \multimap (q \otimes (p \otimes q))) \mathbin{\&} (p \otimes q),$$

which is provable in linear sequent calculus. We must construct a resolution proof of

$$\{!(p \otimes q)\} \vdash_R (q \multimap (q \otimes (p \otimes q))) \mathbin{\&} (p \otimes q).$$

Adopting our lazy approach, we obtain the following proto-proof:

$$\dfrac{\dfrac{\dfrac{\{!(p\otimes q),q\}\vdash^*_R q \qquad \dfrac{\{!(p\otimes q),q\}\vdash^*_R p \quad \{!(p\otimes q),q\}\vdash^*_R q}{\{!(p\otimes q),q\}\vdash^*_R p\otimes q}\otimes}{\{!(p\otimes q),q\}\vdash^*_R q\otimes(p\otimes q)}\otimes}{\{!(p\otimes q)\}\vdash^*_R q\multimap(q\otimes(p\otimes q))}\multimap \qquad \dfrac{\{!(p\otimes q)\}\vdash^\sharp_R p \quad \{!(p\otimes q)\}\vdash^\sharp_R q}{\{!(p\otimes q)\}\vdash^\sharp_R p\otimes q}\otimes}{\{!(p\otimes q)\}\vdash^{*\sharp}_R (q\multimap(q\otimes(p\otimes q)))\mathbin{\&}(p\otimes q)}\mathbin{\&}$$

There are two paths in this proof, marked distinctly by * and $\sharp$. It follows immediately that the expansion $\{p, q\}$ is compatible with the path marked by $\sharp$. The path marked by * is a little more complicated, as the antecedent is a multiset of *mixed* clauses. Thus we must first use up the purely linear part of the antecedent, namely q, before considering the the existence of a suitable expansion of $!(p\otimes q)$. Suppose then that this q is used in the leftmost leaf, then it remains for us to check the existence of an expansion that is compatible with the remaining leaves, with q deleted from the antecedent, *i.e.*, $\{!(p \otimes q)\} \vdash^*_R p$ and $\{!(p \otimes q)\} \vdash^*_R q$. Here we see that the expansion $\{p, q\}$ is compatible with these leaves, so that we can conclude that this proto-proof does indeed determine a resolution proof of the given endsequent.

EXAMPLE 3 Now we introduce the resolution rule. Suppose that we are faced with the endsequent

$$p, q, r, (q \otimes r) \multimap p \vdash p \otimes p,$$

which is provable in linear sequent calculus. Since we have that $[p, q, r, (q \otimes r) \multimap p] = \{p, q, r, (q \otimes r) \multimap p\}$, we must construct a resolution proof of

$$\{p, q, r, (q \otimes r) \multimap p\} \vdash_R p \otimes p.$$

Adopting our lazy approach we obtain the following proto-proof:

$$\dfrac{\{p,q,r,(q\otimes r)\multimap p\}\vdash^*_R p \qquad \dfrac{\{p,q,r,(q\otimes r)\multimap p\}\vdash^*_R p \qquad \dfrac{\{p,q,r,(q\otimes r)\multimap p\}\vdash^*_R q \quad \{p,q,r,(q\otimes r)\multimap p\}\vdash^*_R r}{\{p,q,r,(q\otimes r)\multimap p\}\vdash^*_R q\otimes r}\otimes}{\{p,q,r,(q\otimes r)\multimap p\}\vdash^*_R p}\,\mathrm{res}}{\{p,q,r,(q\otimes r)\multimap p\}\vdash^*_R p\otimes p}\otimes$$

We construct paths as before (in this case there is just one, marked by *), but upon reaching the lazy resolution rule (labelled by res) we must take adjust our calculations. The branch of the lazy version of the resolution rule that corresponds to the branch of the resolution rule of the form $\{\,L\,\}\vdash_R L$, in this case $\{p,q,r,(q\otimes r)\multimap p\}\vdash^*_R p$, is not included in the calculation of the usage of the elements of the antecedent; however the resolvant clause, in this case the clause $(q\otimes r)\multimap p$, is used at this point in the proof. We note that for this proto-proof, each of the remaining components, namely p,q,r, is used exactly once in the remaining leaves; and so we conclude that a resolution proof of the given endsequent is indeed determined.

EXAMPLE 4 Now we introduce the universal quantifier, $\bigwedge$, into the antecedent. Suppose that we are faced with the endsequent

$$\bigwedge x\,.\,(p(x)\otimes q(x))\vdash p(t)\otimes q(u)\,,$$

which is not provable in linear sequent calculus. Since we have that $[\bigwedge x\,.\,(p(x)\otimes q(x))]=\{\,p(x),q(x)\,\}$, with x marked as being global, we must attempt to construct a resolution proof of

$$\{\,p(x),q(x)\,\}\vdash_R p(t)\otimes q(u)\,,$$

and *fail*. Adopting our lazy approach, we obtain the following proto-proof:

$$\dfrac{\{\,p(x),q(x)\,\}\vdash^*_R p(t) \qquad \{\,p(x),q(x)\,\}\vdash^*_R q(u)}{\{\,p(x),q(x)\,\}\vdash^*_R p(t)\otimes q(u)}$$

Each of the leaves (on the unique path marked by *) is of the required form for axioms, but on one branch the global variable x receives the instantiation t and on the other it receives the instantiation u; since these are incompatible, we conclude that the given proto-proof does not determine a resolution proof of the given endsequent.

The examples given above illustrate most of the difficulties encountered in the construction of an interpreter for our notion of linear logic programming. We note that all of the necessary path constructions can be performed dynamically as the search proceeds. However, a few further issues are noteworthy.

The occurrence of & in a program may result in behaviour which appears somewhat disjunctive. For example, as $\phi\vdash\phi$ is provable, it follows that $\phi\,\&\,\psi\vdash\phi$ is provable for any ψ. Thus when searching for a proof of a given goal Δ from a program of the form $\Gamma, \phi \mathbin{\&} \psi$ we need to search for a proof of both $\Gamma, \phi \vdash \Delta$ and $\Gamma, \psi \vdash \Delta$, and halt the search when one branch of the search process succeeds. As the definition of $[\mathcal{P}]$ shows, this may be thought of as finding a formula amongst the alternatives for which the desired result follows. Hence, rather than literally constructing each alternative program, we may think of $\&$ as a mutual exclusion operator; if any one of the formulae C_i in $C_1 \mathbin{\&} \ldots \mathbin{\&} C_n$ has been used in the proof, then none of the others may be used. There may be several such C_j for which the goal succeeds, but only one such formula may be used in a given proof. For example, both $p(a) \vdash \bigvee x \,.\, p(x)$ and $p(b) \vdash \bigvee x \,.\, p(x)$ are provable, and so is $p(a) \mathbin{\&} p(b) \vdash \bigvee x \,.\, p(x)$.

For these reasons an abstract machine for this linear logic programming language will need to incorporate labelling and mutual exclusion facilities in addition to the usual unification and resolution mechanisms. Another possible feature of an implementation is to incorporate modules into the abstract machine along the lines of [20], as mentioned in Section 1. Such a modules system relies on the presence of goal formulae of the form $D \multimap G$, and as such formulae may be present in the body of a clause, it is possible that several formulae may be added to the program during computation, which corresponds to allowing several different modules to be used.

Another complication presented by the linear language which is not present in the intuitionistic one[22] is the management of the goal. In the case of intuitionistic logic, the goal consists of only one formula at any given stage, and so the only degree of freedom is involved with the management of the different branches of the search process. Typically this is done by a depth-first strategy for efficiency reasons. In our case there is an extra degree of freedom as the goal may consist of several formulae at any one time. It would seem natural to bow to the usual practice and reduce each formula in a depth-first manner, but it is less clear whether or not to apply the same strategy to the *choice* of formula. This question is further complicated due to the fact that clauses may be deleted from the program and that programs and goals may be split into smaller progams and goals during the computation process. Given these complications, it might be best to reduce all formulae in the goal to atoms before proceeding with any resolution step. It is anticipated that experimentation with a prototype implementation will provide some insight into the feasibility of various strategies.

8 Conclusion and Further Work

We have given a proof-theoretic basis for logic programming in linear logic, and we have seen how a fragment of linear logic may be used as a logic programming language. As described in Section 6, hereditary Harrop formulae (and hence Horn clauses) may be encoded into the linear system in such a way that the properties of the intuitionistic system are preserved, and so our system may be seen as a generalization of the intuitionistic one. In particular, "mixed" deduction may be performed, using either the linear resolution

[22]As presented in [20].

rule or the standard resolution rule. This provides the the programmer with a facility to specify an upper bound on the number of times that a given clause may be used during execution; if the clause C is to be used twice only during the execution of the goal $\mathcal{G}$, then this may be expressed by attempting to prove $C, C \vdash \mathcal{G}$. Thus our language may be used to incorporate both bounded and unbounded resources in this sense.

Linear logic has been applied to the study of concurrency, and hence it is perhaps not surprising that labelling and mutual exclusion mechanisms will form an important part of the relevant abstract machine. This suggests that the above linear logic programming language may provide an appropriate setting for the study of concurrent logic programming. The "resolve and retract" nature of the linear resolution rule seems particularly apt in this context. The mutual exclusion properties of the connectives $\otimes$ and $\&$ also indicate the concurrent nature of the language, as well as being prime candidates for the application of parallel execution techniques. The use of $\otimes$ as a search operator seems to be a valuable feature in its own right, and is the subject of active research.

A prototype implementation is under way, and it is hoped that experiments with this implementation will lead to the development of appropriate search strategies and management techniques. There has been an increasing interest in linear logic amongst the logic programming community of late, particularly in regard to object-oriented programming [2]. We aim to investigate applications of linear logic programming, particularly in connection with recent work of Abramsky [1].

It was shown in Section 5 how the quantale semantics for linear logic may be adapted to the linear logic programming language given in this paper. However, this semantics is not entirely satisfactory, and we plan to investigate a categorical semantics for linear logic programming. Some possibilities in this area are the categorical semantics of Asperti [4] and Corradini and Montanari [9] for classical logic programming.

An interesting point to note is that in [25] there is a semantics for non-commutative linear logic, *i.e.,* where $\phi \otimes \psi$ need not be equivalent to $\psi \otimes \phi$. This clearly has potential for the study of fixed computation rules, such as *left-to-right* with *depth-first search.*

Another area of future research is to investigate *intermediate linear logic.* The restricted linear logic presented in this paper is not full classical linear logic: negation of literals only is permitted. However, we certainly have more than intuitionistic linear logic, due to the fact that succedents may be multisets of formulae, and not just singleton sets. We aim we develop a theory of intermediate linear logic in the manner of Fitting [11] for modal and intuitionistic logics.

A related matter is the occurrence of ! in a goal. In the language studied in this paper, the only place that ! could occur was in a clause, and not (positively) in a goal. This is due to the failure of the relevant permutation results. It is possible to allow $!G$ under some circumstances: such as in a language with fewer constructs or a more restricted class of definite formulae than in this paper. The reader is referred to [17] for a discussion of such an alternative. It is hoped that a study of intermediate linear logic may be helpful in identifying such cases.

9 Acknowledgements

The authors are grateful to Simon Ambler, Gianluigi Bellin, Dale Miller (who pointed out an error), Gordon Plotkin and Lincoln Wallen for their interest in, and encouragement of, this work.

10 References

[1] Abramsky, S. *Computational Interpretations of Linear Logic.* Research Report DOC 90/20, Imperial College of Science, Technology and Medicine, October 1990.
[2] Andreoli, J-M., Pareschi, R. *Linear Objects: Logical Processes with Built-in Inheritance.* Proceedings of the International Conference on Logic Programming 496-510, Jerusalem, June, 1990.
[3] Andrews, P.B. *Theorem-proving via general matings.* J. Assoc. Comp. Mach. 28(2):193-214, 1981.
[4] Asperti, A. *Categorical Topics in Computer Science.* Dottorato Di Ricerca in Informatica, Università di Pisa – Genova – Udine, 1990.
[5] Avron, A. *Simple Consequence Relations.* LFCS report, ECS-LFCS-87-30. University of Edinburgh, 1987.
[6] Avron, A. *Foundations and Proof Theory of 3-valued Logics.* LFCS report, ECS-LFCS-88-48. University of Edinburgh, 1988.
[7] Bibel, W. *Computationally Improved Versions of Herbrand's Theorem.* Logic Colloquium '81, pp. 11-28. North-Holland, 1982.
[8] Brown, C., Gurr, D. *A Categorical Linear Framework for Petri Nets.* Proc. 5th Annual IEEE Symposium on Logic in Computer Science. Philadelphia, June 1990. IEEE Computer Society Press.
[9] Corradini, A., Montanari, U. *An Algebraic Semantics for Logic Programs.* Proc. 1990 North American Conference on Logic Programming. MIT Press, 1990.
[10] Curry, H.B. *The permutability of rules in the classical inferential calculus.* J. Symb. Log. 17, pp. 245-248, 1952.
[11] Fitting, M. *Proof Methods for Modal and Intuitionistic Logics.* D. Reidel, 1983.
[12] Gentzen, G. *Untersuchungen über das logische Schliessen.* Mathematische Zeitschrift 39(1934), pp. 176-210, 405-431.
[13] Gehlot, V., Gunter, C. *Normal Process Representatives.* Proc. 5th Annual IEEE Symposium on Logic in Computer Science.. Philadelphia, June 1990. IEEE Computer Society Press.
[14] Girard, J-Y. *Linear Logic.* Theor. Comp. Sci. 50, 1987, pp. 1-102.
[15] Girard, J-Y., Lafont, Y., Taylor, P. *Proofs and Types.* Cambridge University Press, 1989.
[16] Harland, J.A. *A Proof-Theoretic Analysis of Logic Programming.* Technical Report 90/21, Department of Computer Science, University of Melbourne.

[17] Hodas, J., Miller, D. *Logic Programming in a Fragment of Intuitionistic Linear Logic: Extended Abstract.* Draft, University of Edinburgh, December 1990.
[18] Kleene, S.C. *Mathematical Logic.* Wiley and Sons, 1968.
[19] Lloyd, J. *Foundations of Logic Programming.* Springer-Verlag, 1984.
[20] Miller, D. *A Logical Analysis of Modules in Logic Programming.* J. Log. Prog. 6(1&2), 1989, pp. 79-108.
[21] Miller, D., Nadathur, G., Pfenning, F., Ščedrov, A. *Uniform Proofs as a Foundation for Logic Programming.* University of Pennsylvania report, MS-CIS-89-36. To appear in the Annals of Pure and Applied Logic.
[22] Pym, D.J. *Proofs, Search and Computation in General Logic.* Ph.D thesis. University of Edinburgh, Department of Computer Science, CST-69-90, 1990. (Also published as LFCS report, ECS-LFCS-90-125.)
[23] Pym, D.J., Wallen L.A. *Investigations into Proof-search in a System of First-order Dependent Function Types.* Proc. 10th International Conference on Automated Deduction, Kaiserslautern FRG, July 1990. Lecture Notes in Artificial Intelligence 449. Springer-Verlag, 1990.
[24] Wallen, L.A. *Automated Deduction in Non-classical Logics.* MIT Press, 1989.
[25] Yetter, D. *Quantales and (Non-commutative) Linear Logic.* J. Symb. Log. 55(1), 1990, pp. 41-64.

A Linear Sequent Calculus

We present the two-sided linear sequent calculus. We let ϕ and ψ range over formulae and Γ and Δ, *etc.* range over sets of formulae.

$$\frac{}{\phi \vdash \phi}\ \text{axiom}$$

$$\frac{\Gamma \vdash \phi, \Delta \quad \Gamma', \phi \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \text{cut}$$

$$\frac{\Gamma, \phi, \psi, \Gamma' \vdash \Delta}{\Gamma, \psi, \phi, \Gamma' \vdash \Delta}\ \text{X-L} \qquad \frac{\Gamma \vdash \Delta, \phi, \psi, \Delta'}{\Gamma \vdash \Delta, \psi, \phi, \Delta'}\ \text{X-R}$$

$$\frac{\Gamma \vdash \phi, \Delta}{\Gamma, \phi^{\perp} \vdash \Delta}\ \perp\text{-L} \qquad \frac{\Gamma, \phi \vdash \Delta}{\Gamma \vdash \phi^{\perp}, \Delta}\ \perp\text{-R}$$

$$\frac{\Gamma, \phi, \psi \vdash \Delta}{\Gamma, \phi \otimes \psi \vdash \Delta}\ \otimes\text{-L} \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma' \vdash \psi, \Delta'}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes\text{-R}$$

$$\frac{\Gamma, \phi \vdash \Delta}{\Gamma, \phi \,\&\, \psi \vdash \Delta} \qquad \frac{\Gamma, \psi \vdash \Delta}{\Gamma, \phi \,\&\, \psi \vdash \Delta}\ \&\text{-L} \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \,\&\, \psi, \Delta}\ \&\text{-R}$$

$$\frac{\Gamma, \phi \vdash \Delta \quad \Gamma, \psi \vdash \Delta}{\Gamma, \phi \oplus \psi \vdash \Delta}\ \oplus\text{-L} \qquad \frac{\Gamma \vdash \phi, \Delta}{\Gamma \vdash \phi \oplus \psi, \Delta} \qquad \frac{\Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \oplus \psi, \Delta}\ \oplus\text{-R}$$

$$\frac{\Gamma, \phi \vdash \Delta \quad \Gamma', \psi \vdash \Delta'}{\Gamma, \Gamma', \phi \parr \psi \vdash \Delta, \Delta'}\ \parr\text{-L} \qquad \frac{\Gamma \vdash \phi, \psi, \Delta}{\Gamma \vdash \phi \parr \psi, \Delta}\ \parr\text{-R}$$

$$\frac{\Gamma \vdash \phi, \Delta \quad \Gamma', \psi \vdash \Delta'}{\Gamma, \Gamma', \phi \multimap \psi \vdash \Delta, \Delta'}\ \multimap\text{-L} \qquad \frac{\Gamma, \phi \vdash \psi, \Delta}{\Gamma \vdash \phi \multimap \psi, \Delta}\ \multimap\text{-R}$$

$$\frac{\Gamma, \phi \vdash \Delta}{\Gamma, !\phi \vdash \Delta}\ !\text{-L} \qquad \frac{!\Gamma \vdash \phi, ?\Delta}{!\Gamma \vdash !\phi, ?\Delta}\ !\text{-R}$$

$$\frac{!\Gamma, \phi \vdash ?\Delta}{!\Gamma, ?\phi \vdash ?\Delta}\ ?\text{-L} \qquad \frac{\Gamma \vdash \phi, \Delta}{\Gamma \vdash ?\phi, \Delta}\ ?\text{-R}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, !\phi \vdash \Delta}\ W?\text{-L} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash ?\phi, \Delta}\ W?\text{-R}$$

$$\frac{\Gamma, !\phi, !\phi \vdash \Delta}{\Gamma, !\phi \vdash \Delta}\ C?\text{-L} \qquad \frac{\Gamma \vdash ?\phi, ?\phi, \Delta}{\Gamma \vdash ?\phi, \Delta}\ C?\text{-R}$$

$$\frac{\Gamma, \phi[t/x] \vdash \Delta}{\Gamma, \bigwedge x \,.\, \phi \vdash \Delta}\ \bigwedge\text{-L} \qquad \frac{\Gamma \vdash \phi[y/x], \Delta}{\Gamma \vdash \bigwedge x \,.\, \phi, \Delta}\ \bigwedge\text{-R}$$

$$\frac{\Gamma, \phi[y/x] \vdash \Delta}{\Gamma, \bigvee x \,.\, \phi \vdash \Delta}\ \bigvee\text{-L} \qquad \frac{\Gamma \vdash \phi[t/x], \Delta}{\Gamma \vdash \bigvee x \,.\, \phi, \Delta}\ \bigvee\text{-R}$$

where x is not free in Γ, Δ.

We note that the presentation of linear sequent calculus for the full language of linear logic can be simplified by replacing each two-sided sequent

$$\phi_1, \ldots, \phi_m \vdash \psi_1, \ldots, \psi_n$$

by the one-sided sequent

$$\vdash \phi_1^{\perp}, \ldots, \phi_m^{\perp}, \psi_1, \ldots, \psi_n \,.$$

The one-sided rules are then given by

$$\frac{}{\vdash \phi^{\perp}, \phi} \qquad \frac{\vdash \phi, \Delta \quad \vdash \phi^{\perp}, \Delta'}{\vdash \Delta, \Delta'}\ \text{cut} \qquad \frac{\vdash \Delta, \phi, \psi, \Delta'}{\vdash \Delta, \psi, \phi, \Delta'}\ \text{X}$$

$$\frac{\vdash \phi, \Delta \quad \vdash \psi, \Delta'}{\vdash \phi \otimes \psi, \Delta, \Delta'}\ \otimes \qquad \frac{\vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\vdash \phi \,\&\, \psi, \Delta}\ \&$$

$$\frac{\vdash \phi, \Delta}{\vdash \phi \oplus \psi, \Delta} \qquad \frac{\vdash \psi, \Delta}{\vdash \phi \oplus \psi, \Delta}\ \oplus$$

$$\frac{\vdash \phi, \psi, \Delta}{\vdash \phi \maltese \psi, \Delta}\ \maltese$$

$$\frac{\vdash \phi, ?\Delta}{\vdash !\phi, ?\Delta}\ ! \qquad \frac{\vdash \Delta}{\vdash ?\phi, \Delta}\ W? \qquad \frac{\vdash ?\phi, ?\phi, \Delta}{\vdash ?\phi, \Delta}\ C?$$

$$\frac{\vdash \phi[y/x], \Delta}{\vdash \bigwedge x \,.\, \phi, \Delta}\ \bigwedge \qquad \frac{\vdash \phi[t/x], \Delta}{\vdash \bigvee x \,.\, \phi, \Delta}\ \bigvee$$

where x is not free in Δ.

Here linear implication $\multimap$ is defined by $\phi \multimap \psi =_{\mathrm{def}} (\phi \otimes \psi^{\perp})^{\perp}$.

B Quantale Semantics

We present the quantale semantics of linear logic. In the first instance we work with the language without the exponentials ! and ?. We remark that this semantics can be extended to include the exponentials ! and ?. The reader is referred to [25] for the details of this work. We note that this semantics can be extended to include a quite general analysis of exponentials [25].

DEFINITION B.1 (QUANTALES) *A* quantale $(\mathbf{Q}, \bigvee, \otimes)$ *is a complete lattice* $(\mathbf{Q}, \bigvee)$ *equipped with an associative tensor product* $\otimes$ *which distributes over arbitrary* $\bigvee$*s on both sides.* $\mathbf{Q}$ *is said to be* commutative *if* $\otimes$ *is commutative.* $\mathbf{Q}$ *is said to be* unital *if there is an element* $\mathbf{1}$ *such that for all* $a \in \mathbf{Q}$, $a \otimes \mathbf{1} = a = \mathbf{1} \otimes a$. $\square$

Let $\mathbf{Q}$ be a quantale. If we regard $\mathbf{Q}$ as a category in the usual way, via the lattice-order $\leq$, then we obtain:

PROPOSITION B.2 *The endofunctors* $(a \otimes -)$ *and* $(- \otimes a)$ *have right adjoints, denoted by* $\multimap_r$ *and* $\multimap_l$ *respectively.* $\square$

DEFINITION B.3 (DUALIZING ELEMENTS) *An element* d *of a quantale* $\mathbf{Q}$ *is* dualizing *if we have*

$$(a \multimap_r d) \multimap_l d = a = (a \multimap_l d) \multimap_r d$$

for all $a \in \mathbf{Q}$. $\square$

DEFINITION B.4 (COMMUTATIVE GIRARD QUANTALES) *A* commutative Girard quantale *is a commutative quantale equipped with a dualizing element* $\perp$. *The operation* $- \multimap_r \perp (= - \multimap_l \perp)$ *is called* linear negation *and is denoted* $-^{\perp}$. $\square$

DEFINITION B.5 (INTERPRETATION) *An* interpretation *of a linear predicate language $\mathcal{L}$ is an ordered triple $(\mathbf{A}, \mathcal{V}, |-|_{\mathbf{A}})$, where $\mathbf{A}$ is an algebra for the theory with constants $\mathbf{0}$, $\mathbf{1}$, $\top$, $\perp$, a unary operation $^{\perp}$, binary operations $\otimes$, $⅋$, & and $\oplus$, and infinitary operations $\bigvee$ and $\bigwedge$. $\mathcal{V}$ is a subset of $\mathbf{A}$ called the* valid *elements and $|-|_{\mathbf{A}}$ is a map from $\mathcal{F}(\mathcal{L})$ to $\mathbf{A}$.* □

DEFINITION B.6 (QUANTALE INTERPRETATIONS) *A* quantale interpretation *of a linear predicate language $\mathcal{L}$ is an ordered pair $(\mathbf{Q}, |-|_{\mathbf{Q}})$, where $\mathbf{Q}$ is a commutative Girard Quantale and $|-|_{\mathbf{Q}}$ is a map $|-|_{\mathbf{Q}} : \mathcal{A}(\mathcal{L}) \longrightarrow \mathbf{Q}$. An element $q \in \mathbf{Q}$ is* valid *if $\mathbf{1} \leq q$.*
□

The constants and operations required by Definition B.6 are defined by extending those of the quantale by the definition $q_1 ⅋ q_2 =_{\text{def}} (q_1^{\perp} \otimes q_2^{\perp})^{\perp}$. We note that using these operations on $\mathbf{Q}$ there is an obvious extension of $|-|_{\mathbf{Q}}$ from $\mathcal{A}(\mathcal{L})$ to $\mathcal{F}(\mathcal{L})$.

A *semantics* for a linear predicate language is a class of interpretations. A semantics is *sound* with respect to linear sequent calculus if whenever $\vdash \phi_1, \ldots, \phi_m$ is provable $|\phi_1 ⅋ \ldots ⅋ \phi_m|_{\mathbf{I}}$ is valid for every interpretation $\mathbf{I}$ in the semantics. A semantics is *complete* with respect to linear sequent calculus if the validity of $|\phi|_{\mathbf{I}}$ for all interpretations $\mathbf{I}$ in the semantics implies that $\vdash \phi$ is provable in linear sequent calculus.

The reader is referred to [14] and [25] for the proofs of the next two theorems. We remark that it is sufficient to consider one-sided sequents, [14], [15], [25]. We give the construction of the linear term quantale and the quantale interpretation of linear logic in the term quantale, which facilitate the completeness theorem.

THEOREM B.7 (SOUNDNESS OF QUANTALE SEMANTICS) *Quantale semantics is sound with respect to linear sequent calculus: if $\vdash \phi_1, \ldots, \phi_m$ is provable in linear sequent calculus and if $|-|_{\mathbf{Q}}$ is a quantale interpretation then $|\phi_1 ⅋ \ldots ⅋ \phi_m|_{\mathbf{Q}}$ is valid.* □

DEFINITION B.8 (THE LINEAR TERM QUANTALE) *Let $\mathbf{M}$ be the monoid of lists of elements of $\mathcal{F}(\mathcal{L})$. The* linear term quantale *is the quantale $\mathbf{P}(\mathbf{M})$, with elements the subsets of $\mathbf{M}$, with $\bigvee$ as set-theoretic union, $\otimes$ given by*

$$\Phi \otimes \Psi = \{\, \phi\psi \mid \phi \in \Phi, \psi \in \Psi \,\}$$

and $\perp$ given by $\perp = \{\, \Phi \mid \vdash \Phi \text{ is provable} \,\}$, where "provable" means provable in linear sequent calculus. □

It follows that this construction yields a commutative Girard quantale [25].

We now define a quantale interpretation $|-|_{\mathbf{P(M)}}$ of linear logic in the linear term quantale $\mathbf{P}(\mathbf{M})$. Let $\Pr : \mathcal{F}(\mathcal{L}) \longrightarrow \mathbf{P}(\mathbf{M})$ be map defined by

$$\Pr(\phi) =_{\text{def}} \{\, \Phi \mid \vdash \phi, \Phi \text{ is provable} \,\},$$

where "provable" means provable in linear sequent calculus.

DEFINITION B.9 (THE INTERPRETATION $|-|_{\mathbf{P(M)}}$) *We define the quantale interpretation*

$$|-|_{\mathbf{P(M)}} : \mathcal{A} \longrightarrow \mathbf{P(M)}$$

of linear logic in the term quantale $\mathbf{P(M)}$ *by*

$$|-|_{\mathbf{P(M)}} =_{\mathrm{def}} \mathrm{Pr}\restriction\mathcal{A}$$

where $\restriction$ *is the usual restriction of mappings.* □

THEOREM B.10 (COMPLETENESS OF QUANTALE SEMANTICS) *Quantale semantics is complete with respect to linear sequent calculus: if* $|\phi_1 \maltese \ldots \maltese \phi_n|_{\mathbf{P(M)}}$ *is valid then* $\vdash \phi_1, \ldots, \phi_n$ *is provable in linear sequent calculus.* □