Resource-distribution via Boolean constraints

JAMES HARLAND
RMIT University
DAVID PYM
University of Bath

We consider the problem of searching for proofs in sequential presentations of logics with multiplicative (or intensional) connectives. Specifically, we start with the multiplicative fragment of linear logic and extend, on the one hand, to linear logic with its additives and, on the other, to the additives of the logic of bunched implications, **BI**. We give an algebraic method for calculating the distribution of the side-formulæ in multiplicative rules which allows the occurrence or non-occurrence of a formula on a branch of a proof to be determined once sufficient information is available. Each formula in the conclusion of such a rule is assigned a Boolean expression. As a search proceeds, a set of Boolean constraint equations is generated. We show that a solution to such a set of equations determines a proof corresponding to the given search. We explain a range of strategies, from the lazy to the eager, for solving sets of constraint equations. We indicate how to apply our methods systematically to large family of relevant systems.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic**]: Proof Theory; I.2.3 [**Deduction and Theorem Proving**]: Deduction; Logic Programming; I.2.8 [**Problem Solving, Control Methods and Search**]: Graph and tree search strategies

General Terms: Languages, Theory

Additional Key Words and Phrases: algebras, Boolean constraints, proof-search, relevant logics, sequent calculus, substructural logics

1. INTRODUCTION

Proof-search in logics, such as linear logic (LL) [Girard 1987] or the logic of bunched implications (**BI**) [O'Hearn and Pym 1999; Pym 1999; 2002], which have multiplicative connectives requires a mechanism by which the distribution of formulæ, sometimes viewed as *resources*, between different branches of a proof may be calculated. Such mechanisms are usually specified by intricate rules of inference which are used to keep track of the relevant information [Hodas and Miller 1994; Winikoff and Harland 1995; Cervesato et al. 2000; Armelín and Pym 2001] and are tailored for one particular distribution mechanism.

We show how a careful use of Boolean expressions admits an explicit expression

James Harland, School of Computer Science and Information Technology, RMIT University, GPO Box 2476V, Melbourne, 3001, Australia. Email: jah@cs.rmit.edu.au.
David Pym, Department of Computer Science, University of Bath, Claverton Down, Bath BA2 7AY, England, U.K. Email: pym@cs.bath.ac.uk.

of the resource-distribution problem (unlike in the standard sequent calculus for such logics) without making a commitment to a particular strategy for managing the distribution (unlike the tailored inference systems of [Hodas and Miller 1994; Winikoff and Harland 1995]). Thus we pursue an *algebraic* approach to this particular problem of proof-search, allowing a separation of the specification of the problem from the method of its solution. We stress that our concern is with the distribution of formulæ across the multiplicative branches of proofs and, in particular, that we do *not* address the issue of which rules are used in which order in order to find proofs. It follows that our work is essentially orthogonal to studies such as those of Andreoli [Andreoli 1992; 2001], on focusing, and somewhat more related (though indirectly) to Galmiche's [Galmiche 2000] proof-nets construction mechanism.[1] The most closely related work of which we are aware is that of Galmiche, Méry and Pym [Galmiche and Méry 2001a; 2001b; Galmiche et al. 2001], on semantic tableaux for **BI**, in which the resource-distribution problem is handled via calculations using labels drawn from an algebra of Kripke worlds. Our contribution should be seen as rendition of the proof theories of the logics concerned within which the distribution problem is represented explicitly but which provides no analysis of the choice of rules used in the construction of (putative) proof trees.

The key step in our approach is to assign a Boolean expression to each of the side-formulæ in multiplicative rules (*i.e.*, just those formulæ which require distribution). Constraints on the possible values of this expression are then generated during the search process and propagated up the search tree, resulting in a set of Boolean constraint equations. A successful proof-search will generate a solution to this set of equations and hence the corresponding distribution of formulæ across the multiplicative branches of the proof may be calculated. We stress that we do not analyse the relationship between the process by which the constraints are solved and the proof-search process. Such an analysis, which is beyond our present scope, would be substantial and a significant step towards a systematic basis for implementation.

Our remarks above on the nature of our contribution notwithstanding, we show how a range of different strategies for solving the Boolean constraints may be considered in this framework, whilst maintaining the same tree of inference rules. So our algebraic approach allows the separation of the inference process from the particular distribution of formulæ across the branches. To illustrate the distinction, consider the $\otimes$R and $⅋$L rules of linear logic [Girard 1987] below.

$$\frac{\Gamma_1 \vdash \phi_1, \Delta_1 \quad \Gamma_2 \vdash \phi_2, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \phi_1 \otimes \phi_2, \Delta_1, \Delta_2}\ \otimes\mathrm{R} \qquad \text{and} \qquad \frac{\Gamma_1, \phi_1 \vdash \Delta_1 \quad \Gamma_2, \phi_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2, \phi_1 ⅋ \phi_2 \vdash \Delta_1, \Delta_2}\ ⅋\mathrm{L},$$

When considered as rules of deductive inference (proofs are generated from premisses to conclusion), there is no distribution problem; the multisets of formulæ Γ_1, Γ_2, Δ_1 and Δ_2 are known in advance. However, when considered as rules of proof-search (*i.e.*, reductive inference), then these four multisets are unknown and must be calculated in the course of proof-search.

In §2, we address this problem for *multiplicative linear logic*, **MLL**. In particular,

[1] A study of the relationship between our methods and (the construction of) proof nets is intriguing but beyond our present scope.

we show how the problem of distributing the side-formulæ may be addressed by attaching a Boolean expression to each such formula together with some simple rules for the generation of constraints on the value of this expression. The problem of allocation of the formula to a branch is thus reduced to determining (in an appropriate manner) a solution to a set of equations over Boolean expressions. We show that this method is sound and complete for this logic.

In §3, we extend the results of the previous section to include the additive and exponential connectives of linear logic, *i.e.*, to *propositional linear logic*, **PLL**. This is quite straightforward.

In §4, we extend the approach of §2 to the (propositional) *logic of bunched implications*, **BI** [O'Hearn and Pym 1999; Pym 1999; 2002; Pym et al. 2000]. This may be understood as a different extension of **MLL** in which the additive connectives, including a semantically adequate implication, are combined freely with the multiplicatives via the notion of bunches [Read 1988; O'Hearn and Pym 1999; Pym 1999; 2002]. The results we obtain for **BI** are correspondingly more subtle.

In §5, we discuss the manner in which the equations generated by the inference rules may be systematically collected and solved. There are three main classes of strategy here: *lazy*, *eager* and *intermediate*. Lazy strategies are those in which one branch of the proof-tree is selected and then followed to completion;the corresponding equations are solved and then the solution which has been found is broadcast to all other branches. Another branch is then selected and the process continues. Such strategies have properties similar to *depth-first search* and are typically used in the implementation of linear logic programming languages such as Lygon [Harland et al. 1996; Winikoff and Harland 1995; Pym and Harland 1994] and Lolli [Hodas and Miller 1994]. Eager strategies are those in which all branches are searched in parallel and one single set of equations is collected. This single set of equations is then solved once all equations have been found. Intermediate strategies solve a fixed number of branches' worth of equations at one time. We discuss how each of these classes of strategies may be utilized in our framework.

Again, we stress that we do not analyse the relationship between the construction of proofs and the solution of the generated constraints. Such an analysis would be a substantial topic in its own right. Moreover, we suggest that this problem should properly be considered in the context of a study of the relationship between the strategies used to find the shape of a (putative) proof tree and the generation of constraints which determine the distribution of resources. Such a study should probably be executed in the context of a mechanical implementation.

In § 6, we conclude with an account of the application of our methods to the family of relevant logics, as described systematically by Read [Read 1988].

We emphasize that we consider our approach to be an algebraic view of (the resource distribution problem in) proof-search; in particular, this approach may be considered as a form of *labelled deduction* [Gabbay 1996]. Our approach is similar in spirit to that of *proof nets* [Girard 1987] and has its origins in the notion of *path* introduced in [Pym and Harland 1994]; however, it should be noted that in our case we recover multiplicative consequences from additive rules by the use of Boolean constraints, rather than recover additive consequences from an underlying multiplicative system. We have suggested that it is helpful to consider the formulæ to be distributed as resources and, indeed, in the case of linear logic there is a very simple sense in which this is possible: LL can be considered to count the number of occurrences of a formula. The treatment of resources provided by **BI**, on the other hand, is more delicate. This point is discussed further in § 4.

Some previous results on linear logic were presented in [Harland and Pym 1997]; in the present paper, we develop these results further and extend them to **BI**.

2. MULTIPLICATIVES

We begin by restricting our attention to *multiplicative linear logic* **MLL**; this is a fragment common to both linear logic and **BI**. In addition, throughout the paper we consider only propositional fragments. Quantifiers pose no problem *per se* and, especially in the case of linear logic, may be incorporated into what follows in a straightforward manner.[2] However, the essential issues are more simply described in their absence.

Definition 2.1. The formulæ of propositional *multiplicative linear logic*, **MLL**, are as follows:

$$p \mid \phi \otimes \phi \mid \phi \parr \phi \mid \phi \multimap \phi \mid \phi^{\perp} \mid \mathbf{1} \mid \perp$$

where p and, subsequently, q, r, *etc.*, denote propositional letters.

Note that we do not include the exponentials here; further discussion on this point may be found in §3.

In order to represent the appropriate constraints on the distribution of formulæ, we do not require arbitrary Boolean expressions. In particular, we require only those given by the following grammar:

$$e ::= x \mid \overline{x} \mid x.e \mid \overline{x}.e$$

where x is a Boolean variable and the operation . is Boolean multiplication.

However, the results of this paper are not dependent on the precise class of formulæ and so any class of Boolean expressions which includes those above will suffice.[3]

Definition 2.2. An *annotated* formula is a formula ϕ together with a Boolean expression e, denoted as $\phi[e]$. We denote by $exp(\phi)$ the Boolean expression associated with the annotated formula ϕ. A sequent consisting entirely of annotated formulæ is known as a *resource sequent.*

In general, the state of the knowledge of the distribution of the formulæ is characterized by the state of knowledge of the Boolean variables, with the distribution of the formula known iff the corresponding Boolean expression has been assigned a value. Hence in addition to the proof tree, we maintain an assignment of the Boolean variables which appear in the proof tree.

[2]Note that quantifiers for linear logic were included in the earlier paper [Harland and Pym 1997].
[3]Naturally such considerations are vital when the complexity of determining the satisfiability of such expressions is concerned. However, this is a topic beyond the scope of the present paper.

Definition 2.3. Given a multiset of annotated formulæ $\Delta = \{\phi_1[e_1], \ldots, \phi_n[e_n]\}$ and a total assignment I of the Boolean variables in Δ, we define

$$\Delta[I] = \{\phi_1[v_1], \ldots, \phi_n[v_n]\},$$

where e_i has the value v_i under I. We denote by $\Delta[I]^1$ the multiset of annotated formulæ $\phi[e]$ in $\Delta[I]$ such that e evaluates to 1 under I.

We will often identify an unannotated formula ϕ with the annotated formula $\phi[1]$ (for instance, in Proposition 2.9); it will always be possible to disambiguate such annotations from the context.

Definition 2.4. Let $V = \{x_1, x_2, \ldots, x_n\}$ be a set of Boolean variables. Then we denote by $\overline{V}$ the set of Boolean expressions $\{\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}\}$. We denote by $\{e\}^n$ the multiset which contains n copies of the Boolean expression e.

Definition 2.5. Let $\Gamma = \{\phi_1[e_1], \phi_2[e_2], \ldots, \phi_n[e_n]\}$ be a multiset of annotated formulæ and let $\{x_1, x_2, \ldots, x_n\}$ be a set of Boolean variables not occurring in Γ. Then we define

$$\Gamma.\{x_1, x_2, \ldots, x_n\} = \{\phi_1[e_1.x_1], \phi_2[e_2.x_2], \ldots, \phi_n[e_n.x_n]\}.$$

We are now in a position to define the inference rules. Roughly speaking, the Boolean constraints are introduced by the rules which require multiplicative distribution of formulæ ($\otimes$R, $⅋$L, $\multimap$L), are resolved by rules with no premisses (Axiom, $\perp$L, $\mathbf{1}$R) and are maintained by all the others. Note also that the principal formula of each rule must be assigned the value 1; hence the selection of the principal formula also makes some contribution to the solution of the equations.

Definition 2.6. We define the following *sequent calculus with constraints* for **MLL**:

$$\text{Ax} \quad \frac{e_1 = e_2 = 1 \;\; \forall e_3 \in \exp(\Gamma \cup \Delta)\; e_3 = 0}{\Gamma, p[e_1] \vdash p[e_2], \Delta}$$

$$\perp\text{L} \quad \frac{\forall e_2 \in \exp(\Gamma \cup \Delta)\; e_2 = 0}{\Gamma, \perp[e_1] \vdash \Delta} \quad e_1 = 1 \qquad \perp\text{R} \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \perp[e], \Delta} \quad e = 1$$

$$\mathbf{1}\text{L} \quad \frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1}[e] \vdash \Delta} \quad e = 1 \qquad \mathbf{1}\text{R} \quad \frac{\forall e_2 \in \exp(\Gamma \cup \Delta)\; e_2 = 0}{\Gamma \vdash \mathbf{1}[e_1], \Delta} \quad e_1 = 1$$

$$⅋\text{L} \quad \frac{\Gamma.V, \phi_1[e] \vdash \Delta.W \quad \Gamma.\overline{V}, \phi_2[e] \vdash \Delta.\overline{W}}{\Gamma, (\phi_1 ⅋ \phi_2)[e] \vdash \Delta} \quad e = 1 \qquad ⅋\text{R} \quad \frac{\Gamma \vdash \phi_1[e], \phi_2[e], \Delta}{\Gamma \vdash (\phi_1 ⅋ \phi_2)[e], \Delta} \quad e = 1$$

$$\otimes\text{L} \quad \frac{\Gamma, \phi_1[e], \phi_2[e] \vdash \Delta}{\Gamma, (\phi_1 \otimes \phi_2)[e] \vdash \Delta} \quad e = 1 \qquad \otimes\text{R} \quad \frac{\Gamma.V \vdash \phi_1[e], \Delta.W \quad \Gamma.\overline{V} \vdash \phi_2[e], \Delta.\overline{W}}{\Gamma \vdash (\phi_1 \otimes \phi_2)[e], \Delta} \quad e = 1$$

$$\multimap\text{L} \quad \frac{\Gamma.V \vdash \phi_1[e], \Delta.W \qquad \Gamma.\overline{V}, \phi_2[e] \vdash \Delta.\overline{W}}{\Gamma, (\phi_1 \multimap \phi_2)[e] \vdash \Delta} \quad e = 1 \qquad \multimap\text{R} \quad \frac{\Gamma, \phi_1[e] \vdash \phi_2[e], \Delta}{\Gamma \vdash (\phi_1 \multimap \phi_2)[e], \Delta} \quad e = 1$$

$$\text{L}^\perp \quad \frac{\Gamma \vdash \phi[e], \Delta}{\Gamma, \phi^\perp[e] \vdash \Delta} \quad e = 1 \qquad \text{R}^\perp \quad \frac{\Gamma, \phi[e] \vdash \Delta}{\Gamma \vdash \phi^\perp[e], \Delta} \quad e = 1$$

where the rules $\otimes$R, $\parr$L and $\multimap$L have the side-condition that V and W are disjoint sets of Boolean variables, none of which occur in Γ, Δ, ϕ_1 or ϕ_2. We refer to the rules Axiom, $\mathbf{1}$R and $\bot$L as *leaf* rules.

We then require a proof in the above system to consist of not just the appropriate proof tree but also an assignment of the Boolean variables which occur in it.

Definition 2.7. A *resource derivation* is a tree regulated by the rules of the resource calculus in which each formula of the endsequent is assigned a distinct Boolean variable, together with a partial assignment of the Boolean variables appearing in the derivation.

A resource derivation is *total* if its assignment of the Boolean variables is total. Otherwise, the resource derivation is *partial.* A resource derivation is *closed* if all of the leaves of the proof tree are instances of one of the leaf rules. A *resource proof* is a total, closed resource derivation in which all the Boolean variables in the endsequent and all principal formulæ are assigned the value 1.

For notational simplicity, a lack of annotation in any of the rules of the resource calculus implies that the constraints currently applicable to the formula are not changed.

Given a resource derivation, it is then straightforward to recover the corresponding proof tree in linear logic. Note that when the resource derivation is a resource proof, this proof tree will actually be a proof in the linear sequent calculus.

Definition 2.8. Let R be a total resource derivation, with proof tree T and Boolean assignment I. The *linear proof tree corresponding to R under I* is the proof tree obtained by deleting from T all formulæ whose Boolean expression evaluates to 0 under I and any remaining constraints.

There may be many resource proofs which have the same corresponding linear proof — for example, the linear proof corresponding to a resource proof of $\Gamma \vdash \Delta$ will be the same as one corresponding to $\Gamma, \phi[0] \vdash \Delta$ (see Lemma 2.10).

Note that resource proofs do not alter the applicability of the inference rules. In particular, they *do not* allow us to construct a "proof" of $p \otimes q \vdash p \otimes q$ in which the right-hand tensor is reduced first: such a proof is prohibited by the requirement of the $\otimes$L rule that each of p, q and $p \otimes q$ be assigned the same value.

To see this, consider the attempt below at such a resource proof:

$$\dfrac{\dfrac{p[x.z], q[x.z] \vdash p[y]}{(p \otimes q)[x.z] \vdash p[y]}\,\otimes\mathrm{L} \qquad \dfrac{p[x.\overline{z}], q[x.\overline{z}] \vdash q[y]}{(p \otimes q)[x.\overline{z}] \vdash q[y]}\,\otimes\mathrm{L}}{(p \otimes q)[x] \vdash (p \otimes q)[y]}\,\otimes\mathrm{R}$$

Clearly we require that $x = 1$ and $y = 1$ in order for the rules to be applied but it is impossible to make either leaf of this tree into an instance of the Axiom rule because of the impossibility of finding an appropriate value for z.

Resource proofs are sound and complete with respect to **MLL** proofs. As usual, soundness reduces to showing that our global conditions are strong enough to recover proofs from the locally unsound system.

PROPOSITION 2.9 SOUNDNESS OF RESOURCE PROOFS. *Let $\Gamma \vdash \Delta$ be a resource sequent in* **MLL**. *If $\Gamma \vdash \Delta$ has a resource proof R with Boolean assignment I, then the linear proof tree corresponding to R is a linear proof of $\Gamma[I]^1 \vdash \Delta[I]^1$.*

PROOF. By induction on the structure of resource proofs. Assume that $\Gamma \vdash \Delta$ has a resource proof.

In the base case, the sequent is the conclusion of either the Axiom, $\perp$L or **1**R rules. As these cases are all similar, we give the argument for the Axiom case only.

Axiom: We have $\Gamma, p[e_1] \vdash p[e_2], \Delta$ where $e_1 = e_2 = 1$ and $\forall e_3 \in \exp(\Gamma \cup \Delta) e_3 = 0$. Let this assignment of Boolean variables be I. Then $\Gamma[I]^1, (p[e_1])[I]^1 \vdash (p[e_2])[I]^1, \Delta[I]^1$ is just $p \vdash p$ and, clearly, the linear proof tree corresponding to R is a proof of this sequent.

Hence we assume that the result holds for all provable sequents whose resource proof is no more than a given size.

Consider the last rule used in the proof. There are ten cases, many of which are similar, so we give the argument for just the $\otimes$R, $\otimes$L, $\multimap$L and $\multimap$R rules:

$\otimes$L: In this case, we have that $\Gamma = (\phi_1 \otimes \phi_2)[e], \Gamma'$, where $e = 1$, and there is a resource proof of $\Gamma', \phi_1, \phi_2 \vdash \Delta$. By the hypothesis, there is a linear proof of $\Gamma'[I]^1, \phi_1[I]^1, \phi_2[I]^1 \vdash \Delta[I]^1$ for some Boolean assignment I and so there is a linear proof of $\Gamma'[I]^1(\phi_1 \otimes \phi_2)[I]^1 \vdash \Delta[I]^1$, which is just $\Gamma'[I]^1 \vdash \Delta[I]^1$, as required.

$\otimes$R: In this case, we have that $\Delta = (\phi_1 \otimes \phi_2)[e], \Delta'$, where $e = 1$, and $\Gamma.V \vdash \phi_1, \Delta'.W$ and $\Gamma.\overline{V} \vdash \phi_2, \Delta'.\overline{W}$ both have resource proofs for some disjoint sets of Boolean variables V and W. By the hypothesis, we have that there are linear proofs of $(\Gamma.V)[I]^1 \vdash (\phi_1)[I]^1, (\Delta'.W)[I]^1$ and $(\Gamma.\overline{V})[I]^1 \vdash (\phi_2)[I]^1, (\Delta'.\overline{W})[I]^1$ (recall that I must be an assignment of *all* Boolean variables in the proof) and so there is a linear proof of

$$(\Gamma.V)[I]^1, (\Gamma.\overline{V})[I]^1 \vdash (\phi_1 \otimes \phi_2)[I]^1, (\Delta'.W)[I]^1, (\Delta'.\overline{W})[I]^1,$$

which is just $\Gamma[I]^1 \vdash \Delta[I]^1$, as required.

$\multimap$L: In this case, we have that $\Gamma = (\phi_1 \multimap \phi_2)[e1], \Gamma'$, where $e = 1$, and $\Gamma'.V \vdash \phi_1, \Delta.W$ and $\Gamma'.\overline{V}, \phi_2 \vdash \Delta.\overline{W}$ both have resource proofs for some disjoint sets of Boolean variables V and W. By the hypothesis, we have that there are linear proofs of $(\Gamma'.V)[I]^1 \vdash (\phi_1)[I]^1, (\Delta.W)[I]^1$ and $(\Gamma'.\overline{V})[I]^1, (\phi_2)[I]^1 \vdash (\Delta.\overline{W})[I]^1$ (recall that I must be an assignment of *all* Boolean variables in the proof) and so there is a linear proof of

$$(\Gamma'.V)[I]^1, (\Gamma'.\overline{V})[I]^1(\phi_1 \multimap \phi_2)[I]^1 \vdash (\Delta.W)[I]^1, (\Delta.\overline{W})[I]^1,$$

which is just $\Gamma[I]^1 \vdash \Delta[I]^1$, as required.

$\multimap$R: In this case, we have that $\Delta = (\phi_1 \multimap \phi_2)[e], \Delta'$, where $e = 1$, and there is a resource proof of $\Gamma, \phi_1 \vdash \phi_2, \Delta'$. By the hypothesis, there is a linear proof of $\Gamma[I]^1, \phi_1[I]^1 \vdash \phi_2[I]^1, \Delta'[I]^1$, for some Boolean assignment I, and so there is a linear proof of $\Gamma[I]^1 \vdash (\phi_1 \multimap \phi_2)[I]^1, \Delta'[I]^1$, which is just $\Gamma[I]^1 \vdash \Delta[I]^1$, as required.

□

In order to show the completeness of resource proofs, we require the following simple lemma:

LEMMA 2.10. *Let* $\Gamma \vdash \Delta$ *be a resource sequent in* **MLL**. *If* $\Gamma \vdash \Delta$ *has a closed resource derivation* R, *then* $\Gamma, \phi[0] \vdash \Delta$ *and* $\Gamma \vdash \phi[0], \Delta$ *also have closed resource derivations* R_1 *and* R_2 *respectively. Moreover, for any total assignment* I *of the Boolean variables in* R, *the linear proof tree corresponding to* R *under* I *is the same as the linear proof trees corresponding to* R_1 *and* R_2 *respectively under* I.

PROOF. By induction on the structure of resource derivations. Assume that $\Gamma \vdash \Delta$ has a closed resource derivation.

In the base case, the sequent is the conclusion of either the Axiom, $\perp$L or **1**R rules and it is clear that the addition of $\phi[0]$ to either the antecedent or the succedent of $\Gamma \vdash \Delta$ will result in a closed resource derivation of the appropriate sequent. Clearly the linear proof tree property is satisfied in these cases.

Hence we assume that the result holds for all provable sequents whose resource proof is no more than a given size.

Consider the last rule used in the proof. There are 10 cases, of which we give the argument for just $\otimes$L and $\otimes$R; the others are similar.

$\otimes$L: In this case, we have that $\Gamma = (\phi_1 \otimes \phi_2)[e], \Gamma'$, where $e = 1$, and there is a closed resource derivation of $\Gamma', \phi_1[1], \phi_2[1] \vdash \Delta$. By the hypothesis, there are closed resource derivations of both $\Gamma', \phi_1[1], \phi_2[1], \phi[0] \vdash \Delta$ and $\Gamma', \phi_1[1], \phi_2[1] \vdash \Delta, \phi[0]$, so there are closed resource derivations of $\Gamma', (\phi_1 \otimes \phi_2)[1], \phi[0] \vdash \Delta$ and $\Gamma', (\phi_1 \otimes \phi_2)[1] \vdash \Delta, \phi[0]$, as required.
Clearly the linear proof tree property is satisfied in this case.

$\otimes$R: In this case, we have that $\Delta = (\phi_1 \otimes \phi_2)[e], \Delta'$, where $e = 1$, and $\Gamma.V \vdash \phi_1[1], \Delta'.W$ and $\Gamma.\overline{V} \vdash \phi_2[1], \Delta'.\overline{W}$ both have closed resource derivations for some disjoint sets of Boolean variables V and W. By the hypothesis, there are closed resource derivations of both $\Gamma.V, \phi[0] \vdash \phi_1[1], \Delta'.W$ and $\Gamma.\overline{V}, \phi[0] \vdash \phi_2[1], \Delta'.\overline{W}$ and of both $\Gamma.V \vdash \phi_1[1], \phi[0], \Delta'.W$ and $\Gamma.\overline{V} \vdash \phi_2[1], \phi[0], \Delta'.\overline{W}$. Hence there are disjoint sets of variables V' and W' such that $V \subset V'$ and $W \subset W'$ such that $(\Gamma \cup \phi[0]).V' \vdash \phi_1[1], \Delta'.W$ and $(\Gamma \cup \phi[0]).\overline{V'} \vdash \phi_2[1], \Delta'.\overline{W}$ both have closed resource derivations, as do $\Gamma.V \vdash \phi_1[1], (\Delta' \cup \phi[0]).W'$ and $\Gamma.\overline{V} \vdash \phi_2[1], (\Delta' \cup \phi[0]).\overline{W'}$. Thus we have closed resource derivations of both $\Gamma, \phi[0] \vdash (\phi_1 \otimes \phi_2)[1], \Delta'$ and $\Gamma \vdash (\phi_1 \otimes \phi_2)[1], \Delta', \phi[0]$, as required.
Clearly the linear proof tree property is satisfied in this case.

□

We are now in a position to show the completeness of resource proofs for **MLL**.

PROPOSITION 2.11 COMPLETENESS OF RESOURCE PROOFS. *Let* $\Gamma \vdash \Delta$ *be a sequent in* **MLL**. *If* $\Gamma \vdash \Delta$ *has a proof* Φ *in* **MLL**, *then there are disjoint sets of Boolean variables* V *and* W *such that* $\Gamma.V \vdash \Delta.W$ *has a resource proof* R *and the linear proof tree corresponding to* R *is* Φ.

PROOF. By induction on the structure of resource proofs. Assume that $\Gamma \vdash \Delta$ has a resource proof.

In the base case, the sequent is the conclusion of either the Axiom, $\bot$L or **1**R rules. As these are all simialr, we give the argument for the Axiom rule only.

Axiom: We have $p \vdash p$, so clearly there is a resource proof of $p[x] \vdash p[y]$ and the linear proof corresponding to this resource proof is Φ.

Hence we assume that the result holds for all provable sequents whose resource proof is no more than a given size.

Consider the last rule used in the proof. There are ten cases, many of which are similar; so we give the argument for just $\otimes$R, $\otimes$L, $\multimap$L and $\multimap$R.

$\otimes$L: In this case, we have that $\Gamma = \phi_1 \otimes \phi_2, \Gamma'$ and there is a proof of $\Gamma', \phi_1, \phi_2 \vdash \Delta$ which is a subproof of Φ. By the hypothesis, there are disjoint sets of variables V and W such that there is a resource proof of $\Gamma'.V, \phi_1[1], \phi_2[1] \vdash \Delta.W$; so there is a resource proof of $\Gamma'.V, (\phi_1 \otimes \phi_2)[1] \vdash \Delta.W$ and clearly the linear proof corresponding to this resource proof is Φ.

$\otimes$R: In this case, we have that $\Gamma = \Gamma_1, \Gamma_2$, $\Delta = \phi_1 \otimes \phi_2, \Delta_1, \Delta_2$ such that $\Gamma_1 \vdash \phi_1, \Delta_1$ and $\Gamma_2 \vdash \phi_2, \Delta_2$ both have proofs in the linear sequent calculus which are subproofs of Φ. Hence by the hypothesis there are disjoint sets of Boolean variables V_1, V_2, W_1, W_2 such that $\Gamma_1.V_1 \vdash \phi_1[1], \Delta_1.W_1$ and $\Gamma_2.V_2 \vdash \phi_2[1], \Delta_2.W_2$ have resource proofs, (and moreover the linear proofs corresponding to each resource proof is the appropriate subproof of Φ) and so by Lemma 2.10, there are closed resource derivations of $\Gamma_1.V_1, \Gamma_2.\{0\}^n \vdash \phi_1[1], \Delta_1.W_1, \Delta_2.\{0\}^n$ and $\Gamma_1.\{0\}^n, \Gamma_2.V_2 \vdash \phi_2[1], \Delta_1.\{0\}^n, \Delta_2.W_2$. Hence there are new disjoint sets of Boolean variables (*i.e.*, not occurring anywhere in the above two resource sequents) V and W and a total assignment I of $V \cup W$ such that $(\Gamma_1.V_1, \Gamma_2.V_2).V \vdash \phi_1[1], (\Delta_1.W_1, \Delta_2.W_2).W$ and $(\Gamma_1.V_1, \Gamma_2.V_2).\overline{V} \vdash \phi_2[1], (\Delta_1.W_1, \Delta_2.W_2).\overline{W}$ have resource proofs and so there is a resource proof of

$$\Gamma_1.V_1, \Gamma_2.V_2 \vdash (\phi_1 \otimes \phi_2)[1], \Delta_1.W_1, \Delta_2.W_2,$$

i.e., $\Gamma.V' \vdash (\phi_1 \otimes \phi_2)[1], \Delta.W'$ for some disjoint sets of Boolean variables V' and W', and clearly the linear proof corresponding to this resource proof is Φ.

$\multimap$L: In this case, we have that $\Gamma = \phi_1 \multimap \phi_2, \Gamma_1, \Gamma_2$, $\Delta = \Delta_1, \Delta_2$ such that $\Gamma_1 \vdash \phi_1, \Delta_1$ and $\Gamma_2, \phi_2 \vdash \Delta_2$ both have proofs in the linear sequent calculus which are subproofs of Φ. Hence by the hypothesis there are disjoint sets of Boolean variables V_1, V_2, W_1, W_2 such that $\Gamma_1.V_1 \vdash \phi_1[1], \Delta_1.W_1$ and $\Gamma_2.V_2, \phi_2[1] \vdash \Delta_2.W_2$ have resource proofs, (and moreover the linear proofs corresponding to each resource proof is the appropriate subproof of Φ) and so by Lemma 2.10, there are closed resource derivations of $\Gamma_1.V_1, \Gamma_2.\{0\}^n \vdash \phi_1[1], \Delta_1.W_1, \Delta_2.\{0\}^n$ and $\Gamma_1.\{0\}^n, \Gamma_2.V_2, \phi_2[1] \vdash \Delta_1.\{0\}^n, \Delta_2.W_2$. Hence there are new disjoint sets of Boolean variables (*i.e.*, not occurring anywhere in the above two resource sequents) V and W and a total assignment I of $V \cup W$ such that $(\Gamma_1.V_1, \Gamma_2.V_2).V \vdash \phi_1[1], (\Delta_1.W_1, \Delta_2.W_2).W$ and $(\Gamma_1.V_1, \Gamma_2.V_2).\overline{V}, \phi_2[1] \vdash (\Delta_1.W_1, \Delta_2.W_2).\overline{W}$ have resource proofs; so there is a resource proof of $\Gamma_1.V_1, \Gamma_2.V_2, (\phi_1 \multimap \phi_2)[1] \vdash \Delta_1.W_1, \Delta_2.W_2$, *i.e.*, $\Gamma.V', (\phi_1 \multimap \phi_2)[1] \vdash \Delta.W'$, for some disjoint sets of Boolean variables V' and W'; clearly the linear proof corresponding to this resource proof is Φ.

$\multimap$R: In this case, we have that $\Delta = \phi_1 \multimap \phi_2, \Delta'$, and there is a proof of $\Gamma, \phi_1 \vdash \phi_2, \Delta'$ which is a subproof of Φ. By the hypothesis, there are disjoint sets of variables V and W such that there is a resource proof of $\Gamma.V, \phi_1[1] \vdash \phi_2[1], \Delta'.W$; so there is a resource proof of $\Gamma'.V \vdash (\phi_1 \multimap \phi_2)[1], \Delta.W$, and clearly the linear proof corresponding to this resource proof is Φ.

□

The reader will have noted that in the application of any rule, it is always necessary for the expression for the principal formula of the rule to be 1. In the sequel, we omit an explicit statement of this requirement when it is obvious from the context.

3. LINEAR ADDITIVES AND EXPONENTIALS

We now consider how to extend the approach of §2 to include additives and the exponentials of linear logic. As noted above, we consider only propositional fragments in this paper; the quantifiers *per se* provide no great technical problem (indeed, see [Harland and Pym 1997] for a basic treatment).

Hence we now consider the class of formulæ defined in the following definition:

Definition 3.1. The formulæ of *propositional linear logic*, **PLL**, are as follows:

$$p \mid \phi \otimes \phi \mid \phi \parr \phi \mid \phi \multimap \phi \mid \phi \,\&\, \phi \mid \phi \oplus \phi \mid !\phi \mid ?\phi \mid \phi^{\perp} \mid \mathbf{1} \mid \perp \mid \mathbf{0} \mid \top$$

where p and, subsequently, q, r, *etc.*, denote propositional letters.

It is straightforward to extend the rules of resource derivations to this class of formulæ, as below. One point to note is that the Boolean expression technique discussed above may be used to characterize the choice of formula in the $\oplus$R and &L rules. For example, the $\oplus$R rule requires a choice to be made between ϕ_1 and ϕ_2, as below.

$$\oplus\mathrm{R}\ \frac{\Gamma \vdash \phi_i, \Delta}{\Gamma \vdash \phi_1 \oplus \phi_2, \Delta}$$

However, it is clear that by attaching a Boolean variable to ϕ_1 and its complement to ϕ_2 we are able to use the same constraint technique as above. This requires the rule to be reformulated as

$$\oplus\mathrm{R}\ \frac{\Gamma \vdash \phi_1[x], \phi_2[\overline{x}], \Delta}{\Gamma \vdash (\phi_1 \oplus \phi_2)[1], \Delta}$$

Similar remarks apply to the &L rule.

The exponentials are also straightforward, as they do not introduce any distribution problems. It should be noted, though, that the technique of attaching Boolean expressions to formulæ may be adapted to whatever approach is taken to the rules involving exponentials. For example, it is well-known that the weakening rules (W!L and W?R) may be permuted upwards and so may be incorporated into the leaf rules. Also, the rules !L and C!L may be combined into one, as below [Tammet 1994]:

$$\frac{\Gamma, \phi, !\phi \vdash \Delta}{\Gamma, !\phi \vdash \Delta}$$

In either case, it is not hard to see how to adapt the resource versions of these rules to incorporate these adaptations. Hence here we consider only the "naive" exponential rules, as variations of them, as mentioned above, constitute only minor changes to the rules below.

Definition 3.2. We define the following ***sequent calculus with constraints*** for **PLL**:

$$\text{Axiom}\quad \frac{e_1 = e_2 = 1 \;\; \forall e_3 \in \exp(\Gamma \cup \Delta)\; e_3 = 0}{\Gamma, p[e_1] \vdash p[e_2], \Delta}$$

$$\bot\text{L}\quad \frac{\forall e_2 \in \exp(\Gamma \cup \Delta)\; e_2 = 0}{\Gamma, \bot[e_1] \vdash \Delta}\; e_1 = 1 \qquad \bot\text{R}\quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \bot[e], \Delta}\; e = 1$$

$$\mathbf{1}\text{L}\quad \frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1}[e] \vdash \Delta}\; e = 1 \qquad \mathbf{1}\text{R}\quad \frac{\forall e_2 \in \exp(\Gamma \cup \Delta)\; e_2 = 0}{\Gamma \vdash \mathbf{1}[e_1], \Delta}\; e_1 = 1$$

$$\mathbf{0}\text{L}\quad \frac{}{\Gamma, \mathbf{0}[e_1] \vdash \Delta}\; e_1 = 1 \qquad \top\text{R}\quad \frac{e_1 = 1}{\Gamma \vdash \top[e_1]\Delta}$$

$$⅋\text{L}\quad \frac{\Gamma.V, \phi_1[e] \vdash \Delta.W \quad \Gamma.\overline{V}, \phi_2[e] \vdash \Delta.\overline{W}}{\Gamma, (\phi_1 ⅋ \phi_2)[e] \vdash \Delta}\; e = 1 \qquad ⅋\text{R}\quad \frac{\Gamma \vdash \phi_1[e], \phi_2[e], \Delta}{\Gamma \vdash (\phi_1 ⅋ \phi_2)[e], \Delta}\; e = 1$$

$$\otimes\text{L}\quad \frac{\Gamma, \phi_1[e], \phi_2[e] \vdash \Delta}{\Gamma, (\phi_1 \otimes \phi_2)[e] \vdash \Delta}\; e = 1 \qquad \otimes\text{R}\quad \frac{\Gamma.V \vdash \phi_1[e], \Delta.W \quad \Gamma.\overline{V} \vdash \phi_2[e], \Delta.\overline{W}}{\Gamma \vdash (\phi_1 \otimes \phi_2)[e], \Delta}\; e = 1$$

$$\multimap\text{L}\quad \frac{\Gamma.V \vdash \phi_1[e], \Delta.W \qquad \Gamma.\overline{V}, \phi_2[e] \vdash \Delta.\overline{W}}{\Gamma, (\phi_1 \multimap \phi_2)[e] \vdash \Delta}\; e = 1 \qquad \multimap\text{R}\quad \frac{\Gamma, \phi_1[e] \vdash \phi_2[e], \Delta}{\Gamma \vdash (\phi_1 \multimap \phi_2)[e], \Delta}\; e = 1$$

$$\text{L}^{\perp}\quad \frac{\Gamma \vdash \phi[e], \Delta}{\Gamma, \phi^{\perp}[e] \vdash \Delta}\; e = 1 \qquad \text{R}^{\perp}\quad \frac{\Gamma, \phi[e] \vdash \Delta}{\Gamma \vdash \phi^{\perp}[e], \Delta}\; e = 1$$

$$\oplus\text{L}\quad \frac{\Gamma, \phi_1[e] \vdash \Delta \qquad \Gamma, \phi_2[e] \vdash \Delta}{\Gamma, (\phi_1 \oplus \phi_2)[e] \vdash \Delta}\; e = 1 \qquad \oplus\text{R}\quad \frac{\Gamma \vdash \phi_1[x], \phi_2[\overline{x}], \Delta}{\Gamma \vdash (\phi_1 \oplus \phi_2)[e], \Delta}\; e = 1$$

$$\&\text{L}\quad \frac{\Gamma, \phi_1[x], \phi_2[\overline{x}] \vdash \Delta}{\Gamma, (\phi_1 \,\&\, \phi_2)[e] \vdash \Delta}\; e = 1 \qquad \&\text{R}\quad \frac{\Gamma \vdash \phi_1[e], \Delta \quad \Gamma \vdash \phi_2[e], \Delta}{\Gamma \vdash (\phi_1 \,\&\, \phi_2)[e], \Delta}\; e = 1$$

$$!\text{L}\quad \frac{\Gamma, \phi[e] \vdash \Delta}{\Gamma, !\phi[e] \vdash \Delta}\; e = 1 \qquad !\text{R}\quad \frac{!\Gamma \vdash \phi[e], ?\Delta}{!\Gamma \vdash !\phi[e], ?\Delta}\; e = 1$$

$$?\text{L}\quad \frac{!\Gamma, \phi[e] \vdash ?\Delta}{!\Gamma, ?\phi[e] \vdash ?\Delta}\; e = 1 \qquad ?\text{R}\quad \frac{\Gamma \vdash \phi[e], \Delta}{\Gamma \vdash ?\phi[e], \Delta}\; e = 1$$

$$W!\text{L}\quad \frac{\Gamma \vdash \Delta}{\Gamma, !\phi[e] \vdash \Delta}\; e = 1 \qquad W?\text{R}\quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash ?\phi[e], \Delta}\; e = 1$$

$$C!\text{L}\quad \frac{\Gamma, !\phi[e], !\phi[e] \vdash \Delta}{\Gamma, !\phi[e] \vdash \Delta}\; e = 1 \qquad C?\text{R}\quad \frac{\Gamma \vdash ?\phi[e], ?\phi[e], \Delta}{\Gamma \vdash ?\phi[e], \Delta}\; e = 1$$

where the rules $\otimes$R, ⅋L and $\multimap$L have the side-condition that V and W are disjoint sets of Boolean variables, none of which occur in Γ, Δ, ϕ_1 or ϕ_2.

It is then straightforward to show the soundness of resource proofs for **PLL**.

PROPOSITION 3.3 SOUNDNESS OF RESOURCE PROOFS. *Let* $\Gamma \vdash \Delta$ *be a resource sequent in* **PLL**. *If* $\Gamma \vdash \Delta$ *has a resource proof* R *with Boolean assignment* I*, then the linear proof tree corresponding to* R *is a linear proof of* $\Gamma[I]^1 \vdash \Delta[I]^1$.

PROOF. By induction on the structure of resource proofs. Assume that $\Gamma \vdash \Delta$ has a resource proof.

In the base case, the sequent is the conclusion of either the Axiom, $\bot$L, **1**R, **0**L or $\top$R rules and it is clear that the result holds in each of these cases.

Hence we assume that the result holds for all provable sequents whose resource proof is no more than a given size.

Consider the last rule used in the proof. We give the argument just for the cases $\otimes$L, $\otimes$R, &L, &R, C!L; the others are similar.

&L: In this case, we have that $\Gamma = (\phi_1 \,\&\, \phi_2)[1], \Gamma'$ and there is a resource proof of $\Gamma', \phi_1[x], \phi_2[\overline{x}] \vdash \Delta$. By the hypothesis, there is a linear proof of

$$\Gamma'[I]^1, (\phi_1[x])[I]^1, (\phi_2[\overline{x}])[I]^1 \vdash \Delta[I]^1,$$

for some Boolean assignment I, which is clearly either $\Gamma'[I]^1, \phi_1 \vdash \Delta[I]^1$ or $\Gamma'[I]^1, \phi_2 \vdash \Delta[I]^1$, as required.

&R: In this case, we have that $\Delta = (\phi_1 \,\&\, \phi_2)[1], \Delta'$, and $\Gamma \vdash \phi_1[1], \Delta'$ and $\Gamma \vdash \phi_2[1], \Delta'$ both have resource proofs. By the hypothesis, we have that there are linear proofs of $\Gamma[I]^1 \vdash (\phi_1)[I]^1, \Delta'[I]^1$ and $\Gamma[I]^1 \vdash (\phi_2)[I]^1, \Delta'[I]^1$, and so there is a linear proof of $\Gamma[I]^1 \vdash (\phi_1 \,\&\, \phi_2)[I]^1, \Delta'[I]^1$ which is just $\Gamma[I]^1 \vdash \Delta[I]^1$, as required.

$\otimes$L: In this case, we have that $\Gamma = (\phi_1 \otimes \phi_2)[1], \Gamma'$ and there is a resource proof of $\Gamma', \phi_1, \phi_2 \vdash \Delta$. By the hypothesis, there is a linear proof of $\Gamma'[I]^1, \phi_1[I]^1, \phi_2[I]^1 \vdash \Delta[I]^1$, for some Boolean assignment I, and so there is a linear proof of $\Gamma'[I]^1(\phi_1 \otimes \phi_2)[I]^1 \vdash \Delta[I]^1$, which is just $\Gamma'[I]^1 \vdash \Delta[I]^1$, as required.

$\otimes$R: In this case, we have that $\Delta = (\phi_1 \otimes \phi_2)[1], \Delta'$, and $\Gamma.V \vdash \phi_1[1], \Delta'.W$ and $\Gamma.\overline{V} \vdash \phi_2[1], \Delta'.\overline{W}$ both have resource proofs for some disjoint sets of Boolean variables V and W. By the hypothesis, we have that there are linear proofs of $(\Gamma.V)[I]^1 \vdash (\phi_1)[I]^1, (\Delta'.W)[I]^1$ and $(\Gamma.\overline{V})[I]^1 \vdash (\phi_2)[I]^1, (\Delta'.\overline{W})[I]^1$ (recall that I must be an assignment of *all* Boolean variables in the proof), and so there is a linear proof of $(\Gamma.V)[I]^1, (\Gamma.\overline{V})[I]^1 \vdash (\phi_1 \otimes \phi_2)[I]^1, (\Delta'.W)[I]^1, (\Delta'.\overline{W})[I]^1$, which is just $\Gamma[I]^1 \vdash \Delta[I]^1$, as required.

C!L: In this case, we have that $\Gamma = !\phi[1], \Gamma'$ and there is a resource proof of $\Gamma', !\phi[1], !\phi[1] \vdash \Delta$. By the hypothesis, there is a linear proof of

$$\Gamma'[I]^1, !\phi[I]^1, !\phi[I]^1 \vdash \Delta[I]^1,$$

for some Boolean assignment I; hence there is a linear proof of

$$\Gamma'[I]^1, !\phi[I]^1 \vdash \Delta[I]^1,$$

which is just $\Gamma[I]^1 \vdash \Delta[I]^1$, as required.

□

As previously, we require the following lemma in order to show the completeness of resource proofs for **PLL**.

LEMMA 3.4. *Let $\Gamma \vdash \Delta$ be a resource sequent in* **PLL**. *If $\Gamma \vdash \Delta$ has a closed resource derivation R, then $\Gamma, \phi[0] \vdash \Delta$ and $\Gamma \vdash \phi[0], \Delta$ also have closed resource derivations R_1 and R_2 respectively. Moreover, for any total assignment I of the Boolean variables in R, the linear proof tree corresponding to R under I is the same as the linear proof trees corresponding to R_1 and R_2 respectively under I.*

PROOF. The proof is similar to that of Lemma 2.10 and so is omitted. □

We are now in a position to show the completeness of resource proofs for **PLL**.

PROPOSITION 3.5 COMPLETENESS OF RESOURCE PROOFS. *Let $\Gamma \vdash \Delta$ be a sequent in* **PLL**. *If $\Gamma \vdash \Delta$ has a proof Φ in* **PLL**, *then there are disjoint sets of Boolean variables V and W such that $\Gamma.V \vdash \Delta.W$ has a resource proof R and the linear proof tree corresponding to R is Φ.*

PROOF. By induction on the structure of resource proofs. Assume that $\Gamma \vdash \Delta$ has a resource proof.

In the base case, the sequent is the conclusion of either the Axiom, $\bot$L, **1**R, **0**L or $\top$R rules and it is clear that the result holds in each of these cases.

Hence we assume that the result holds for all provable sequents whose resource proof is no more than a given size.

Consider the last rule used in the proof. We give the argument just for the cases $\otimes$L, $\otimes$R, &L, &R, C!L; the others are similar.

$\otimes$L: In this case, we have that $\Gamma = \phi_1 \otimes \phi_2, \Gamma'$ and there is a proof of $\Gamma', \phi_1, \phi_2 \vdash \Delta$ which is a subproof of Φ. By the hypothesis, there are disjoint sets of variables V and W such that there is a resource proof of $\Gamma'.V, \phi_1[1], \phi_2[1] \vdash \Delta.W$ and so there is a resource proof of $\Gamma'.V, (\phi_1 \otimes \phi_2)[1] \vdash \Delta.W$, and clearly the linear proof corresponding to this resource proof is Φ.

$\otimes$R: In this case, we have that $\Gamma = \Gamma_1, \Gamma_2$, $\Delta = \phi_1 \otimes \phi_2, \Delta_1, \Delta_2$ such that $\Gamma_1 \vdash \phi_1, \Delta_1$ and $\Gamma_2 \vdash \phi_2, \Delta_2$ both have proofs in the linear sequent calculus which are subproofs of Φ. Hence by the hypothesis there are disjoint sets of Boolean variables V_1, V_2, W_1, W_2 such that $\Gamma_1.V_1 \vdash \phi_1[1], \Delta_1.W_1$ and $\Gamma_2.V_2 \vdash \phi_2[1], \Delta_2.W_2$ have resource proofs, (and moreover the linear proofs corresponding to each resource proof is the appropriate subproof of Φ) and so by Lemma 3.4, there are closed resource derivations of $\Gamma_1.V_1, \Gamma_2.\{0\}^n \vdash \phi_1[1], \Delta_1.W_1, \Delta_2.\{0\}^n$ and $\Gamma_1.\{0\}^n, \Gamma_2.V_2 \vdash \phi_2[1], \Delta_1.\{0\}^n, \Delta_2.W_2$. Hence there are new disjoint sets of Boolean variables (*i.e.*, not occurring anywhere in the above two resource sequents) V and W and a total assignment I of $V \cup W$ such that $(\Gamma_1.V_1, \Gamma_2.V_2).V \vdash \phi_1[1], (\Delta_1.W_1, \Delta_2.W_2).W$ and $(\Gamma_1.V_1, \Gamma_2.V_2).\overline{V} \vdash \phi_2[1], (\Delta_1.W_1, \Delta_2.W_2).\overline{W}$ have resource proofs and so there is a resource proof of

$$\Gamma_1.V_1, \Gamma_2.V_2 \vdash (\phi_1 \otimes \phi_2)[1], \Delta_1.W_1, \Delta_2.W_2,$$

i.e., $\Gamma.V' \vdash (\phi_1 \otimes \phi_2)[1], \Delta.W'$, for some disjoint sets of Boolean variables V' and W', and clearly the linear proof corresponding to this resource proof is Φ.

&L: In this case, we have that $\Gamma = \phi_1 \,\&\, \phi_2, \Gamma'$ and there is a proof of $\Gamma', \phi_i \vdash \Delta$ which is a subproof of Φ. Without loss of generality, let $i = 1$. By the hypothesis, there are disjoint sets of variables V and W such that there is a resource proof of $\Gamma'.V, \phi_1[1] \vdash \Delta.W$ and by Lemma 3.4 there is a resource proof of $\Gamma'.V, \phi_1[1], \phi_2[0] \vdash \Delta.W$, *i.e.*, there is a resource proof of $\Gamma'.V, \phi_1[x], \phi_2[\overline{x}] \vdash \Delta.W$, giving us a resource proof of $\Gamma'.V, (\phi_1 \,\&\, \phi_2)[1] \vdash \Delta.W$ and so clearly the linear proof corresponding to this resource proof is Φ.

&R: In this case, we have that $\Delta = \phi_1 \,\&\, \phi_2, \Delta'$ such that $\Gamma \vdash \phi_1, \Delta'$ and $\Gamma \vdash \phi_2, \Delta$ both have proofs in the linear sequent calculus which are subproofs of Φ. Hence by the hypothesis there are disjoint sets of Boolean variables V_1, V_2, W_1, W_2 such that $\Gamma.V_1 \vdash \phi_1[1], \Delta.W_1$ and $\Gamma.V_2 \vdash \phi_2[1], \Delta.W_2$ have resource proofs, (and moreover the linear proofs corresponding to each resource proof is the appropriate subproof of Φ). Now, as each of these has the property that the Boolean expression attached to each formula in the conclusion evaluates to 1, we can choose $V_1 = V_2$ and $W_1 = W_2$ and so we have a resource proof of $\Gamma.V_1 \vdash (\phi_1 \,\&\, \phi_2)[1], \Delta.W_1$; clearly, the linear proof corresponding to this resource proof is Φ.

C!L: In this case, we have that $\Gamma = !\phi, \Gamma'$ and there is a proof of $\Gamma', !\phi, !\phi \vdash \Delta$ which is a subproof of Φ. By the hypothesis, there are disjoint sets of variables V and W such that there is a resource proof of $\Gamma'.V, !\phi[1], !\phi[1] \vdash \Delta.W$ and so there is a resource proof of $\Gamma'.V, !\phi[1] \vdash \Delta.W$; clearly, the linear proof corresponding to this resource proof is Φ.

□

Note that the reason that it is straightforward to adapt the $\oplus$R and &L rules as above because the Boolean constraints to be solved are of the same form as those in §2, *i.e.*, determining the satisfiability of expressions such as $x.y.\overline{z}$. This may be thought of as determining the truth of the existentially quantified expression $\exists x.\exists y.\exists z.(x.y.\overline{z})$. As such constraints can contain only existential quantifiers, they form a class of constratins which can be solved comparatively simply. Such considerations of quantified Boolean constraints suggest that the introduction of universally quantified expressions may be useful. In particular, such expressions can be used to provide "slices", as used in proof nets for the additive rules of linear logic [Girard 1994]. Such slices may be thought of as indexing each branch of the proof with a Boolean variable, as the only point of difference between the endsequents of each branch is the formula to which the variable is attached. Thus we arrive at a proof structure which is linear rather than binary, at the expense of a correspondingly more complex class of Boolean constraints to be solved. For example, in such a system the $\otimes$R would become

$$\frac{\Gamma \vdash \phi_1[x], \phi_2[\overline{x}], \Delta}{\Gamma \vdash \phi_1 \otimes \phi_2, \Delta} \otimes \mathrm{R}$$

which would result in a universally quantified expression. Hence the complexity of finding solutions is increased but with the benefit of providing an algebraic interpretation of proofs in which the additives are taken as basic and in which the multiplicatives are introduced via the Boolean constraints. This contrasts with proof nets, in which the additives are introduced to a multiplicative system by means of the Boolean expressions. Further development on this point is beyond the scope of this paper; we will take up this thread in subsequent research.

$$\begin{array}{rcll} \Gamma & ::= & \phi & \text{propositional assumption} \\ & | & \emptyset_m & \text{multiplicative unit} \\ & | & \Gamma, \Gamma & \text{multiplicative combination} \\ & | & \emptyset_a & \text{additive unit} \\ & | & \Gamma ; \Gamma & \text{additive combination} \end{array}$$

Fig. 1. Bunches

4. BUNCHED ADDITIVES

We now turn to a different extension of **MLL**, in which the additives are handle rather differently. Roughly, in the logic of bunched implications, **BI**, there are two implications, one multiplicative and one additive. This necessitates a more complex structure in the antecedent, as we must distinguish additive contexts from multiplicative contexts. However, as we shall see, the same basic technique as above may be applied to the problem of distribution of formulæ across multiplicative branches of a proof.

In **BI**, there are multiplicative versions of not only of conjunction and disjunction (as in linear logic) but also of implication (unlike linear logic).[4] It follows that we have the following two rules for implications in succedents:

$$\to \mathrm{R} \quad \frac{\Gamma; \phi_1 \vdash \phi_2}{\Gamma \vdash \phi_1 \to \phi_2} \qquad -\!\!* \mathrm{R} \quad \frac{\Gamma, \phi_1 \vdash \phi_2}{\Gamma \vdash \phi_1 -\!\!* \phi_2}$$

These rules thus show that implication is intimately associated with the way in which antecedents are formed. In particular, because of the existence of the two implication rules above, it is necessary to have two constructors for antecedents: one multiplicative, written as "," and logically equivalent to **BI**'s multiplicative conjunction, $*$, and one additive, written as ";" and logically equivalent to **BI**'s additive conjunction, $\wedge$.[5]

Thus antedecents in **BI** are not simply finite sequences, as in linear logic, but finite trees, with the leaf nodes being formulæ and the internal nodes denoted by either "," or ";" , and are referred to as *bunches*.[6] The grammar of bunches is given in Figure 1. For example, the bunch $\phi_1, ((\phi_3, \phi_4); \phi_2)$ may be drawn as in Figure 2.

The main point of the definition of bunches is that ";" admits the structural properties of Weakening and Contraction, whereas "," does not: this distinction allows the correct formulation of the two implications. Bunches may be represented using lists of lists, *etc.*, as described in [Read 1988]. We write $\Gamma(\Delta)$, and refer to

[4] Whilst additive implication has been defined for linear logic, see, for example, [Troelstra 1992], their semantic status is, in our opinion [Pym 2002], questionable.

[5] The underlying semantic structure of **BI**'s proofs, which can be represented as the terms of a simply-typed λ-calculus, is given by a *bi-cartesian doubly closed category* [O'Hearn and Pym 1999; Pym 1999; 2002], which carries two closed structures, one symmetric monoidal and one bi-cartesian. There are many examples of such categories [O'Hearn and Pym 1999; Pym 1999; 2002].

[6] A multiple-conclusioned version of **BI**, with disjunctive bunched structure in succedents, may also be formulated [Pym 2002]. For simplcity, we restrict our attention here to the single-conclusioned case.

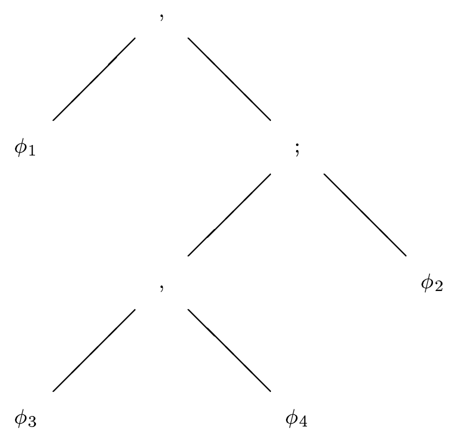

Fig. 2. An example bunch: $\phi_1, ((\phi_3, \phi_4); \phi_2)$

Δ as a *sub-bunch* of Γ, for a bunch Γ in which Δ appears as a sub-tree and write $\Gamma[\Delta'/\Delta]$ for Γ with Δ replaced by Δ'. We write $\Gamma(-)$ to denote a bunch Γ which is incomplete and which may be completed by placing a bunch in its hole, and will use this notation to refer to that part of $\Gamma(\Delta)$ which is not part of Δ. We require that "," and ";" be commutative monoids, giving rise to the coherent equivalence, $\Gamma \equiv \Gamma'$, as follows:[7]

Coherent Equivalence: $\Gamma \equiv \Gamma'$

1 Commutative monoid equations for $\emptyset_a$ and ";".
2 Commutative monoid equations for $\emptyset_m$ and ",".
3 Congruence: if $\Delta \equiv \Delta'$ then $\Gamma(\Delta) \equiv \Gamma(\Delta')$.

Note that ";" and "," do not distribute over one another. We use $=$ for syntactic identity of bunches.

BI may also be understood semantically [O'Hearn and Pym 1999; Pym 1999; 2002]. The key structure is that of a *preordered monoid*, $\mathcal{M} = (M, \cdot, e, \sqsubseteq)$ which provides the worlds for a Kripke-style forcing semantics. The additives receive a semantics in the usual intuitionistic way, whereas the multiplicatives receive a semantics in the style of Kripke, Urquhart and Pym [Kripke 1965; Urquhart 1972; O'Hearn and Pym 1999; Pym 1999; 2002]:

$$m \models \phi \mathrel{-\!\!*} \psi \text{ iff for all } n \text{ such that } n \models \phi,\ m \cdot n \models \psi$$
$$m \models \phi * \psi \text{ iff there exist } n \text{ and } n', \text{ where } m \sqsubseteq n \cdot n', \text{ such that } n \models \phi \text{ and } n' \models \psi.$$

Categorically, such a semantics can be formulated in doubly closed categories of the form $\mathbf{Set}^C$, where C is a small symmetric monoidal category [O'Hearn and Pym 1999; Pym 1999; 2002]. This semantics may be interpreted as being a model

[7]The multiplicative operation, ",", need not, in general, be commutative and, indeed, such non-commutative versions of **BI** are possible [Pym 2002].

of the notion of *resource*: it may be argued that the following are a basic set of assumptions about resources:

—There should be a *zero* resource; call it e;

—The should be a *combination* of resources; call it $\cdot$;

—The should be a a *comparison* of resources; call it $\sqsubseteq$.

The assumptions lead to the forcing semantics based on preordered monoids. The leading characteristic of this resource semantics is the *sharing interpretation*: we can analyse the sharing and privacy of the resources accessed by co-existing computations. The topics, including soundness and completeness theorems and computationally natural concrete models, are discussed fully in [O'Hearn and Pym 1999; Pym 1999; 2002; Pym et al. 2000]. The logical view of resources sketched here should be contrasted with that which is available for linear logic, in which a proposition is interpreted directly as a resource and its number of uses is counted.

For further information on **BI**, the reader is referred to [O'Hearn and Pym 1999; Pym 1999; 2002]. A good summary of the use of bunched structure in substructural logic may be found in [Restall 2000; Read 1988].

Definition 4.1. The formulæ of the *propositional logic of bunched implications*, **BI**, are as follows:[8]

$$\phi ::= p \mid \top \mid \bot \mid I \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \to \phi \mid \phi * \phi \mid \phi \mathrel{-\!\!*} \phi \,,$$

in which p and, subsequently, q, r, *etc.*, denote propositional letters.

Below is (a version of) the sequent calculus **LBI** for propositional **BI** [O'Hearn and Pym 1999; Pym 1999; 2002].

Definition 4.2. The calculus **LBI** may be defined as follows:

[8]Note that **BI** denotes the unit of additive conjunction as $\top$ whereas linear logic denotes it as $\mathbf{1}$.

$$
\begin{array}{llll}
\text{Axiom} & \dfrac{}{\phi \vdash \phi} & \text{E} & \dfrac{\Gamma \vdash \phi}{\Delta \vdash \phi} \text{ where } \Gamma \equiv \Delta \\[2ex]
\text{W} & \dfrac{\Gamma(\Delta) \vdash \phi}{\Gamma(\Delta;\Delta') \vdash \phi} & \text{C} & \dfrac{\Gamma(\Delta;\Delta) \vdash \phi}{\Gamma(\Delta) \vdash \phi} \\[2ex]
I\text{L} & \dfrac{\Gamma(\emptyset_m) \vdash \phi}{\Gamma(I) \vdash \phi} & I\text{R} & \dfrac{}{\emptyset_m \vdash I} \\[2ex]
\top\text{L} & \dfrac{\Gamma(\emptyset_a) \vdash \phi}{\Gamma(\top) \vdash \phi} & \top\text{R} & \dfrac{}{\emptyset_a \vdash \top} \\[2ex]
\bot\text{L} & \dfrac{}{\bot \vdash \phi} & & \\[2ex]
-\!\!*\,\text{L} & \dfrac{\Gamma \vdash \phi_1 \quad \Delta(\Delta', \phi_2) \vdash \phi}{\Delta(\Delta', \Gamma, \phi_1 -\!\!* \phi_2) \vdash \phi} & -\!\!*\,\text{R} & \dfrac{\Gamma, \phi_1 \vdash \phi_2}{\Gamma \vdash \phi_1 -\!\!* \phi_2} \\[2ex]
*\text{L} & \dfrac{\Gamma(\phi_1, \phi_2) \vdash \phi}{\Gamma(\phi_1 * \phi_2) \vdash \phi} & *\text{R} & \dfrac{\Gamma \vdash \phi_1 \quad \Delta \vdash \phi_2}{\Gamma, \Delta \vdash \phi_1 * \phi_2} \\[2ex]
\to\text{L} & \dfrac{\Gamma \vdash \phi_1 \quad \Delta(\Gamma; \phi_2) \vdash \phi}{\Delta(\Gamma; \phi_1 \to \phi_2) \vdash \phi} & \to\text{R} & \dfrac{\Gamma; \phi_1 \vdash \phi_2}{\Gamma \vdash \phi_1 \to \phi_2} \\[2ex]
\wedge\text{L} & \dfrac{\Gamma(\phi_1; \phi_2) \vdash \phi}{\Gamma(\phi_1 \wedge \phi_2) \vdash \phi} & \wedge\text{R} & \dfrac{\Gamma \vdash \phi_1 \quad \Gamma \vdash \phi_2}{\Gamma \vdash \phi_1 \wedge \phi_2} \\[2ex]
\vee\text{L} & \dfrac{\Gamma(\phi_1) \vdash \phi \quad \Gamma(\phi_2) \vdash \phi}{\Gamma(\phi_1 \vee \phi_2) \vdash \phi} & \vee\text{R} & \dfrac{\Gamma \vdash \phi_i}{\Gamma \vdash \phi_1 \vee \phi_2}
\end{array}
$$

We will refer to the Axiom, IR, $\top$R and $\bot$L rules as *leaf* rules.

Note that, since **LBI** admits Cut-elimination [Pym 2002], we omit the Cut rule. Also, we have used the additive version of the rules $\wedge$R, $\to$L and $\vee$L rules (see [O'Hearn and Pym 1999; Pym 1999; 2002] for details). This is done because the derivability of these rules in **LBI** may be interpreted as showing that these rules need not introduce any new distribution problems. Note also that, as in [O'Hearn and Pym 1999; Pym 1999; 2002], we could have used a similarly additive form of the Axiom rule, *viz.*

$$\text{Axiom}\ \dfrac{}{\phi; \Gamma \vdash \phi}$$

However, we have chosen not to do this as it unnecessarily complicates the structure of the antecedent. Recall that the basic technique is to relax the multiplicative rules, so that "extra" formulæ are present in the sequent, and then the constraints are used to determine the distribution of all formulæ across multiplicative branches. In the linear case, as antecedents are multiplicative, there is no issue. In the **BI** case, the antecedent could be additive, as in the above rule. The resource version of the Axiom rule would then be

$$\text{Axiom}\ \frac{\forall e \in \exp(\Delta)\ e = 0}{(\phi[e_1];\Gamma), \Delta \vdash \phi}\ e_1 = 1,$$

which seems unnecessarily complex, especially as the rules of weakening and contraction are still required. So we use the original form of the Axiom rule but the additive versions of the $\wedge$R, $\rightarrow$L and $\vee$L rules.

Now the rules which require attention are just the leaf rules (Axiom, IR, $\perp$L, 1R) and $-\!*$ L (as $*$R is precisely the same rule as for linear logic). We reformulate the leaf rules as follows:

$$\text{Axiom}\ \frac{\forall e \in \exp(\Delta)\ e = 0}{\phi[e_1], \Delta \vdash \phi}\ e_1 = 1 \qquad \perp\text{L}\ \frac{\forall e \in \exp(\Delta)\ e = 0}{\perp [e_1], \Delta \vdash \phi}\ e_1 = 1$$

$$I\text{R}\ \frac{\forall e \in \exp(\Delta)\ e = 0}{\emptyset_m[e_1], \Delta \vdash I}\ e_1 = 1 \qquad 1\text{R}\ \frac{\forall e \in \exp(\Delta)\ e = 0}{\emptyset_a[e_1], \Delta \vdash 1}\ e_1 = 1$$

Note that as **LBI** is single-conclusioned, there is no need to consider distribution on the right of $\vdash$, and hence the formulæ in the succedents never require a Boolean expression. Otherwise, these are little different from similar rules in linear logic.

The only remaining rule is $-\!*$ L. Here the main point to note is that we need to take account of the bunch structure but otherwise this is handled similarly to the $\multimap$L rule in linear logic. The new form of the rule is then

$$\frac{\Gamma.V \vdash \phi_1 \qquad \Delta(\Gamma.\overline{V}, \phi_2[e]) \vdash \phi}{\Delta(\Gamma, (\phi_1 -\!* \phi_2)[e]) \vdash \phi}\ e = 1,$$

where $\Gamma, \phi_1 -\!* \phi_2$ is the "maximal multiplicative super-bunch" of $\phi_1 -\!* \phi_2$ in $\Delta(\Gamma, \phi_1 -\!* \phi_2)$. For example, given the sequent

$$p, (q; (r, s, t -\!* u)) \vdash v$$

we get the following inference step:

$$\frac{r[x_1], s[x_2] \vdash t \qquad p, (q; (r[\overline{x_1}], s[\overline{x_2}], u[1])) \vdash v}{p, (q; (r, s, (t -\!* u)[1])) \vdash v}$$

Here $\Gamma = \{r, s\}$ and not $\{r\}$ or $\{s\}$. We formalize this notion below.

Definition 4.3. Let $\Delta(\Gamma)$ be a bunch. Then the *maximal multiplicative super-bunch* of Γ in Δ is defined as follows:

—If $\Delta(\Gamma) = \Gamma$ or the parent of Γ in Δ is ";", then the maximal multiplicative super-bunch of Γ in Δ is just Γ;

—If the parent of Γ in Δ is ",", then the maximal multiplicative super-bunch of Γ in Δ is the maximal multiplicative super-bunch of the parent of Γ in Δ.

Note that in this case we have to attach expressions to bunches. For example, consider the provable sequent $r, (s; t) \vdash r * s$. A resource derivation of this sequent would be

$$\dfrac{r[x],(s;t)[y] \vdash r \qquad \dfrac{r[\overline{x}], s[\overline{y}] \vdash s}{r[\overline{x}],(s;t)[\overline{y}] \vdash s}\,\mathrm{W}}{r,(s;t) \vdash r * s}\,{*}\mathrm{R}$$

from which we get $x = 1$, $y = 0$, corresponding to the sequent proof

$$\dfrac{r \vdash r \qquad \dfrac{s \vdash s}{s;t \vdash s}\,\mathrm{W}}{r,(s;t) \vdash r * s}\,{*}\mathrm{R}$$

So we must attach expressions to bunches. However, it will not necessarily be correct to simply attach the expression to every formula in the bunch, as a multiplicative bunch may contain an additive sub-bunch. For example, in the sequent above, we wish to distribute the two sub-bunches r and $(s;t)$ across the two multiplicative branches and not the three sub-bunches r, s and t. So we must define the operation $\Gamma.V$, in which Γ is a bunch so that it respects this multiplicative structure.

Definition 4.4. Let Γ be a bunch and V a set of distinct variables. We define the operation $\Gamma.V$ as follows:

—If $\Gamma = \phi$, then $|V| = 1$ and $\Gamma.V = \phi[x]$;
—If $\Gamma = (\Gamma'; \Delta)$, then $|V| = 1$ and $\Gamma.V = (\Gamma'; \Delta)[x]$;
—If $\Gamma = (\Gamma', \Delta)$, then $\Gamma.V = (\Gamma'.V_1, \Delta.V_2)$, where V_1, V_2 are disjoint sets of variables such that $V = V_1 \cup V_2$.

Thus we have that $r,(s;t).\{x, y\} = r[x],(s;t)[y]$. Note that the linear version of this operation may be thought of as a bunch in which the only constructor is ",". Note also that in the additive implication it is also useful to identify the maximal *additive* super-bunch of the principal formula. Doing so allows us to show that there is indeed no distribution problem arising from the additive form of the $\to$L rule, as then Γ and Δ may easily be determined from the rule application. For example, given the sequent

$$q,(p_3; p_1; p_1 \to p_2) \vdash q * p_2,$$

it would seem appropriate to apply the $\to$L rule as

$$\dfrac{\dfrac{p_1 \vdash p_1}{p_3; p_1 \vdash p_1}\,W \qquad \dfrac{q \vdash q \qquad \dfrac{p_2 \vdash p_2}{p_3; p_1; p_2 \vdash p_2}\,W}{q,(p_3; p_1; p_2) \vdash q * p_2}\,{*R}}{q,(p_3; p_1; p_1 \to p_2) \vdash q * p_2}\,\to \mathrm{L}$$

rather than as

$$\dfrac{p_1 \vdash p_1 \qquad \dfrac{q \vdash q \qquad \dfrac{p_2 \vdash p_2}{p_3; p_1; p_2 \vdash p_2}\,W}{q,(p_3; p_1; p_2) \vdash q * p_2}\,{*R}}{q,(p_3; p_1; p_1 \to p_2) \vdash q * p_2}\,\to \mathrm{L}$$

The former application avoids having to determine where p_3 will be used by sending it to each branch and deleting it (via the W rule) as required. However, we do not pursue this direction in this paper, as it has no effect on the issues of distribution. We get the following rules:

Definition 4.5. We define the following *sequent calculus with constraints* for **BI**:

$$\text{Axiom}\quad \frac{\forall e \in \exp(\Delta)\ e = 0}{\phi[e_1], \Delta \vdash \phi}\ e_1 = 1 \qquad\qquad \text{E}\quad \frac{\Gamma[e] \vdash \phi}{\Delta[e] \vdash \phi}\ e = 1 \text{ where } \Gamma \equiv \Delta$$

$$\text{W}\quad \frac{\Gamma(\Delta[e]) \vdash \phi}{\Gamma((\Delta; \Delta')[e]) \vdash \phi}\ e = 1 \qquad\qquad \text{C}\quad \frac{\Gamma(\Delta[e]; \Delta[e]) \vdash \phi}{\Gamma(\Delta[e]) \vdash \phi}\ e = 1$$

$$I\text{L}\quad \frac{\Gamma(\emptyset_m[e]) \vdash \phi}{\Gamma(I[e]) \vdash \phi}\ e = 1 \qquad\qquad I\text{R}\quad \frac{\forall e \in \exp(\Delta)\ e = 0}{\emptyset_m[e_1], \Delta \vdash I}\ e_1 = 1$$

$$\top\text{L}\quad \frac{\Gamma(\emptyset_a[e]) \vdash \phi}{\Gamma(\top[e]) \vdash \phi}\ e = 1 \qquad\qquad \top\text{R}\quad \frac{\forall e \in \exp(\Delta)\ e = 0}{\emptyset_a[e_1], \Delta \vdash \top}\ e_1 = 1$$

$$\bot\text{L}\quad \frac{\forall e \in \exp(\Delta)\ e = 0}{\bot\ [e_1], \Delta \vdash \phi}\ e_1 = 1$$

$$-\!\!*\text{L}\quad \frac{\Gamma.V \vdash \phi_1 \qquad \Delta(\Gamma.\overline{V}, \phi_2[e]) \vdash \phi)}{\Delta(\Gamma, \phi_1 -\!\!* \phi_2[e]) \vdash \phi}\ e = 1 \qquad\qquad -\!\!*\text{R}\quad \frac{\Gamma, \phi_1[e] \vdash \phi_2}{\Gamma \vdash \phi_1 -\!\!* \phi_2}\ e = 1$$

$$*\text{L}\quad \frac{\Gamma(\phi_1[e], \phi_2[e]) \vdash \phi}{\Gamma(\phi_1 * \phi_2[e]) \vdash \phi}\ e = 1 \qquad\qquad *\text{R}\quad \frac{\Gamma.V \vdash \phi_1 \qquad \Gamma.\overline{V} \vdash \phi_2}{\Gamma \vdash \phi_1 * \phi_2}$$

$$\to\text{L}\quad \frac{\Gamma \vdash \phi_1 \qquad \Delta(\Gamma; \phi_2[e]) \vdash \phi}{\Delta(\Gamma; \phi_1 \to \phi_2[e]) \vdash \phi}\ e = 1 \qquad\qquad \to\text{R}\quad \frac{\Gamma; \phi_1[e] \vdash \phi_2}{\Gamma \vdash \phi_1 \to \phi_2}\ e = 1$$

$$\wedge\text{L}\quad \frac{\Gamma(\phi_1[e]; \phi_2[e]) \vdash \phi}{\Gamma(\phi_1 \wedge \phi_2[e]) \vdash \phi}\ e = 1 \qquad\qquad \wedge\text{R}\quad \frac{\Gamma \vdash \phi_1 \qquad \Gamma \vdash \phi_2}{\Gamma \vdash \phi_1 \wedge \phi_2}$$

$$\vee\text{L}\quad \frac{\Gamma(\phi_1[e]) \vdash \phi \qquad \Gamma(\phi_2[e]) \vdash \phi}{\Gamma(\phi_1 \vee \phi_2[e]) \vdash \phi}\ e = 1 \qquad\qquad \vee\text{R}\quad \frac{\Gamma \vdash \phi_i}{\Gamma \vdash \phi_1 \vee \phi_2}$$

where the $-\!\!*$L rule has the side condition that $\Gamma, \phi_1 -\!\!* \phi_2$ is the maximal multiplicative super-bunch containing $\phi_1 -\!\!* \phi_2$, and the rules $-\!\!*$L and $*$R have the side condition that V is a set of Boolean variables not occurring in Γ, Δ, ϕ_1, ϕ_2 or ϕ.

We are now in a position to define resource derivations for propositional **BI**. This is actually the same definition as Definition 2.7.

Definition 4.6. A *resource derivation* is a tree regulated by the rules of the resource calculus in which each formula of the endsequent is assigned a distinct Boolean variable, together with a (possibly partial) assignment of the Boolean variables appearing in the derivation.

A resource derivation is *total* if its assignment of the Boolean variables is total. Otherwise, the resource derivation is *partial*. A resource derivation is *closed* if all of the leaves of the proof tree are leaf rules. A *resource proof* is a total, closed resource derivation in which all the Boolean variables in the endsequent are assigned the value 1.

As above, it is then straightforward to recover an **LBI** proof from a resource proof.

Definition 4.7. Let R be a total resource derivation, with proof tree T and Boolean assignment I. The **LBI** *proof tree corresponding to* R is the proof tree obtained by deleting from T all formulæ whose Boolean expression evaluates to 0 under I.

For example, consider the resource derivation below.

$$\dfrac{\dfrac{r[x_1], p[x_2] \vdash p}{r[x_1], (p;t)[x_2] \vdash p}\,W \qquad \dfrac{\dfrac{r[\overline{x_1}.y_1], (p;t)[\overline{x_2}.y_2], q[y_3] \vdash q \quad r[\overline{x_1}.\overline{y_1}], (p;t)[\overline{x_2}.\overline{y_2}], q[\overline{y_3}] \vdash r}{r[\overline{x_1}], (p;t)[\overline{x_2}], q \vdash q * r}\,{*}\mathrm{R}}{(r[\overline{x_1}], (p;t)[\overline{x_2}], q); s \vdash q * r}\,W}{(r, (p;t), p \mathrel{-\!\!*} q[1]); s \vdash q * r}\,{-\!\!*}\,\mathrm{L}$$

Solving the equations gives us

$$x_1 = 0, x_2 = 1, y_1 = 0, y_3 = 1$$

which in turn gives us the resource proof

$$\dfrac{\dfrac{r[0], p[1] \vdash p}{r[0], (p;t)[1] \vdash p}\,W \qquad \dfrac{\dfrac{r[0], (p;t)[0], q[1] \vdash q \quad r[1], (p;t)[0], q[0] \vdash r}{r[1], (p;t)[0], q \vdash q * r}\,{*}\mathrm{R}}{(r[1], (p;t)[0], q); s \vdash q * r}\,W}{(r, (p;t), p \mathrel{-\!\!*} q[1]); s \vdash q * r}\,{-\!\!*}\,\mathrm{L}$$

The corresponding **LBI** proof is then

$$\dfrac{\dfrac{p \vdash p}{(p;t) \vdash p}\,W \qquad \dfrac{\dfrac{q \vdash q \quad r \vdash r}{r, q \vdash q * r}\,{*}\mathrm{R}}{(r, q); s \vdash q * r}\,W}{(r, (p;t), p \mathrel{-\!\!*} q); s \vdash q * r}\,{-\!\!*}\,\mathrm{L}$$

Note that this proof could be part of a larger proof, such as the one below, but no new constraints are introduced by the larger context.

$$\dfrac{\dfrac{\dfrac{p \vdash p}{(p;t) \vdash p}\,W \qquad \dfrac{\dfrac{q \vdash q \quad r \vdash r}{r, q \vdash q * r}\,{*}\mathrm{R}}{(r, q); s \vdash q * r}\,W}{(r, (p;t), p \mathrel{-\!\!*} q); s \vdash q * r}\,{-\!\!*}\,\mathrm{L} \qquad \dfrac{s \vdash s}{(r, (p;t), p \mathrel{-\!\!*} q); s \vdash s}\,W}{(r, (p;t), p \mathrel{-\!\!*} q); s \vdash (q * r) \wedge s}\,\wedge\mathrm{R}$$

It should be noted that the bunch structure may also simplify some matters. For example, when two variables share the same expression in a multiplicative context, such as the bunch $r[x], s[x]$, it is not hard to see that this is equivalent to $(r, s)[x]$, which is a simpler representation of the same information. It may happen that this latter structure requires a decomposition because of a need to split this multiplicative bunch, in which case we would revert to the former form (and presumably add extra variables to each expression, *e.g.*, $r[x.y], s[x.z]$). Thus the bunch structure introduces some further dynamics in the "tagging" of formulæ.

It should also be noted that this tagging is transparent to coherent equivalence. In particular, it should be clear that if $\Gamma \equiv \Delta$, then $\Gamma(\Delta'[0]) \equiv \Delta(\Delta'[0])$ and $\Gamma(\Delta'[1]) \equiv \Delta(\Delta'[1])$. It should also be clear that (by abuse of notation) $\Gamma.V(\Delta[0]) = \Gamma(\Delta[0]).V$.

We are now in a position to show the soundness of resource proofs for **BI**.

PROPOSITION 4.8 SOUNDNESS OF RESOURCE PROOFS. *Let $\Gamma \vdash \phi$ be a resource sequent in* **BI**. *If $\Gamma \vdash \phi$ has a resource proof R with Boolean assignment I, then the* **LBI** *proof tree corresponding to R is an* **LBI** *proof of $\Gamma[I]^1 \vdash \phi$.*

PROOF. We proceed by induction on the size of the resource derivation.

In the base case, the resource proof consists of just one of the leaf rules. Hence there are four cases, of which we give the argument for Axiom only, the others being similar.

In this case, the endsequent of the resource proof is just $\phi[e_1], \Delta \vdash \phi$ with $e_1 = 1$ and $\forall e \in \exp(\Delta) e = 0$. Hence $\Gamma[I]^1 = \phi$ and the **LBI** proof corresponding to this resource proof is just $\phi \vdash \phi$.

Hence we assume that the result holds for all proofs of no more than a given size.

There are numerous cases, of which we give the argument just for $-\!\!*$ L, $-\!\!*$ R, $*$L, $*$R and $\rightarrow$L, the others being similar.

- $-\!\!*$ L: In this case, the endsequent is $\Delta(\Gamma, \phi_1 -\!\!* \phi_2[e]) \vdash \phi$, where $e = 1$, and the premisses are $\Gamma.V \vdash \phi_1$ and $\Delta(\Gamma.\overline{V}, \phi_2[e]) \vdash \phi)$. Now as the endsequent has a resource proof, there is a Boolean assignment I of V such that both premisses have resource proofs. So, by the induction hypothesis, we have that the **LBI** proof tree corresponding to the resource derivation of $\Gamma.V \vdash \phi_1$ is an **LBI** proof of $\Gamma.V[I]^1 \vdash \phi_1$, and the **LBI** proof tree corresponding to the resource derivation of $\Delta(\Gamma.\overline{V}, \phi_2[1]) \vdash \phi$ is an **LBI** proof of $\Delta(\Gamma.\overline{V}, \phi_2[1])[I]^1 \vdash \phi_1$. Now as $\Gamma, \phi_1 -\!\!* \phi_2$ is the maximal multiplicative super-bunch of $\phi_1 -\!\!* \phi_2$ in Δ, and as I is a total assignment of Boolean variables, we have that $\Gamma.V[I]^1 = \Gamma_1$ and $\Gamma.\overline{V}[I]^1 = \Gamma_2$, where $\Gamma_1, \Gamma_2 = \Gamma$, and so the **LBI** proof tree corresponding to the resource derivation of $\Delta(\Gamma, \phi_1 -\!\!* \phi_2[1]) \vdash \phi$ is an **LBI** proof of $\Delta(\Gamma, \phi_1 -\!\!* \phi_2[1])[I]^1 \vdash \phi$.
- $-\!\!*$ R: In this case, the endsequent is $\Gamma \vdash \phi_1 -\!\!* \phi_2$ and the premiss is $\Gamma, \phi_1[1] \vdash \phi_2$. So, by the induction hypothesis, we have that the **LBI** proof tree corresponding to the resource derivation of $\Gamma, \phi_1[1] \vdash \phi_2$ is an **LBI** proof of $\Gamma, \phi_1[1][I]^1 \vdash \phi_2$; so the **LBI** proof tree corresponding to the resource derivation of $\Gamma \vdash \phi_1 -\!\!* \phi_2$ is an **LBI** proof of $\Gamma[I]^1 \vdash \phi_1 -\!\!* \phi_2$.
- $*$L: In this case, the endsequent is $\Gamma(\phi_1 * \phi_2[e]) \vdash \phi$, where $e = 1$, and the premiss is $\Gamma(\phi_1[e], \phi_2[e]) \vdash \phi$. So, by the induction hypothesis, we have that the **LBI** proof tree corresponding to the resource derivation of $\Gamma(\phi_1[1], \phi_2[1]) \vdash \phi$ is an **LBI** proof of $\Gamma(\phi_1, \phi_2)[I]^1 \vdash \phi$; hence the **LBI** proof tree corresponding to the resource derivation of $\Gamma(\phi_1 * \phi_2[1]) \vdash \phi$ is an **LBI** proof of $\Gamma(\phi_1 * \phi_2)[I]^1 \vdash \phi$.
- $*$R: In this case, the endsequent is $\Gamma \vdash \phi_1 * \phi_2$ and the premisses are $\Gamma.V \vdash \phi_1$ and $\Gamma.\overline{V} \vdash \phi_2$, so there is a Boolean assignment I such that both premisses

have a resource-proof. Hence by the induction hypothesis we have that the **LBI** proof tree corresponding to the resource derivation of $\Gamma.V \vdash \phi_1$ is an **LBI** proof of $\Gamma.V[I]^1 \vdash \phi_1$, and that the **LBI** proof tree corresponding to the resource derivation of $\Gamma.\overline{V} \vdash \phi_2$ is an **LBI** proof of $\Gamma.\overline{V}[I]^1 \vdash \phi_2$. As I is a total assignment of Boolean variables, we have that $\Gamma.V[I]^1 = \Gamma_1$ and $\Gamma.\overline{V}[I]^1 = \Gamma_2$, where $\Gamma_1, \Gamma_2 = \Gamma$, and so the **LBI** proof tree corresponding to the resource derivation of $\Gamma \vdash \phi_1 * \phi_2$ is an **LBI** proof of $\Gamma \vdash \phi_1 * \phi_2$.

$\rightarrow$L: In this case, the endsequent is $\Gamma(\phi_1 \rightarrow \phi_2[e]) \vdash \phi$, where $e = 1$, and the premisses are $\Gamma \vdash \phi_1$ and $\Gamma(\phi_2[e]) \vdash \phi$. Hence, by the induction hypothesis, we have that the **LBI** proof tree corresponding to the resource derivation of $\Gamma \vdash \phi_1$ is an **LBI** proof of $\Gamma[I]^1 \vdash \phi_1$, and the **LBI** proof tree corresponding to the resource derivation of $\Gamma(\phi_2) \vdash \phi$ is an **LBI** proof of $\Gamma(\phi_2)[I]^1 \vdash \phi$, and so the **LBI** proof tree corresponding to the resource derivation of $\Gamma(\phi_1 \rightarrow \phi_2[1]) \vdash \phi$ is an **LBI** proof of $\Gamma(\phi_1 \rightarrow \phi_2)[I]^1 \vdash \phi$.

□

As for **MLL** and **PLL**, we require the following lemma:

LEMMA 4.9. *Let $\Gamma \vdash \phi$ be a resource sequent in* **BI**. *If $\Gamma \vdash \phi$ has a closed resource derivation, then for any bunch Δ' we have that $\Gamma(\Delta'[0]) \vdash \phi$ also has a closed resource derivation.*

PROOF. We proceed by induction on the size of the resource derivation.

In the base case, the resource derivation consists of just one of the leaf rules. Hence there are four cases, of which we give the argument for Axiom only, the others being similar.

In this case, $\Gamma = \phi, \Delta$ and the endsequent of the resource derivation is just $\phi[e_1], \Delta \vdash \phi$ with $e_1 = 1$ and $\forall e \in \exp(\Delta) e = 0$. Hence it is clear that there is also a closed resource derivation of $\Gamma(\Delta'[0]) \vdash \phi$.

Hence we assume that the result holds for all proofs of no more than a given size. There are numerous cases, of which we give the argument just for $-\!\!*$L, $-\!\!*$R, $*$L, $*$R and $\rightarrow$L, the others being similar.

$-\!\!*$L: In this case, the endsequent is $\Delta(\Gamma, \phi_1 -\!\!* \phi_2[e]) \vdash \phi$, where $e = 1$, and the premisses are $\Gamma.V \vdash \phi_1$ and $\Delta(\Gamma.\overline{V}, \phi_2[e]) \vdash \phi)$. Hence by the hypothesis we have that $\Gamma.V(\Delta'[0]) \vdash \phi_1$ and $\Delta(\Gamma(\Delta'[0]).\overline{V}, \phi_2[e])) \vdash \phi)$ both have closed resource derivations (recall that the position of the bunch Δ' may be arbitrary), and so $\Delta(\Gamma(\Delta'[0]), \phi_1 -\!\!* \phi_2[e]) \vdash \phi$ has a closed resource derivation.

$-\!\!*$R: In this case, the endsequent is $\Gamma \vdash \phi_1 -\!\!* \phi_2$ and the premiss is $\Gamma, \phi_1[e] \vdash \phi_2$, where $e = 1$. Hence, by the induction hypothesis, we have that $\Gamma(\Delta'[0]), \phi_1[e] \vdash \phi_2$ has a closed resource derivation; hence so does $\Gamma(\Delta'[0]) \vdash \phi_1 -\!\!* \phi_2$.

$*$L: In this case, the endsequent is $\Gamma(\phi_1 * \phi_2[e]) \vdash \phi$, where $e = 1$, and the premiss is $\Gamma(\phi_1[e], \phi_2[e]) \vdash \phi$. Hence, by the induction hypothesis, we have that $\Gamma(\phi_1[e], \phi_2[e])(\Delta'[0]) \vdash \phi$ has a closed resource derivation; hence so does $\Gamma(\phi_1[e] * \phi_2[e])(\Delta'[0]) \vdash \phi$.

$*$R: In this case, the endsequent is $\Gamma \vdash \phi_1 * \phi_2$ and the premisses are $\Gamma.V \vdash \phi_1$ and $\Gamma.\overline{V} \vdash \phi_2$. Hence, by the induction hypothesis, we have that $\Gamma.V(\Delta'[0]) \vdash \phi_1$

and $\Gamma.\overline{V}(\Delta'[0]) \vdash \phi_2$ have closed resource derivations, and as $(\Gamma.V)(\Delta'[0]) = \Gamma(\Delta'[0]).V$, so does $\Gamma(\Delta'[0]) \vdash \phi_1 * \phi_2$.

$\rightarrow$L: In this case, the endsequent is $\Gamma(\phi_1 \rightarrow \phi_2[e]) \vdash \phi$, where $e = 1$, and the premisses are $\Gamma \vdash \phi_1$ and $\Gamma(\phi_2[e]) \vdash \phi$. Hence, by the induction hypothesis, we have that $\Gamma(\Delta'[0]) \vdash \phi_1$ and $\Gamma(\phi_2[e])(\Delta'[0]) \vdash \phi$ have closed resource derivations; hence so does $\Gamma(\phi_1 \rightarrow \phi_2[e])(\Delta'[0]) \vdash \phi$.

□

We are now in a position to show the completeness of resource proofs for **BI**.

Proposition 4.10 completeness of resource proofs. *Let $\Gamma \vdash \phi$ be a sequent in* **BI**. *If $\Gamma \vdash \phi$ has a proof Φ in* **LBI**, *then there is a set of Boolean variables V such that $\Gamma.V \vdash \phi$ has a resource proof R and the* **LBI** *proof tree corresponding to R is Φ.*

Proof. We proceed by induction on the height of the **LBI** proof.

In the base case, the rule used is one of Axiom, IR, 1R and $\perp$L. We give the argument for Axiom only, the others being similar:

In this case, the sequent is just $\phi \vdash \phi$ and it is clear that this **LBI** proof corresponds to the resource proof $\dfrac{}{\phi[x] \vdash \phi}$, with $x = 1$.

Hence we assume that the result holds for all proofs of no more than a given size. There are numerous cases, of which we give the argument just for W, $-\!\!*$L, $-\!\!*$R, $*$L, $*$R, $\rightarrow$L, $\rightarrow$R and $\wedge$R, the others being similar.

W: In this case, the conclusion is $\Gamma(\Delta;\Delta') \vdash \phi$ and the premiss is $\Gamma(\Delta) \vdash \phi$ and so, by the hypothesis, there is a resource proof of $\Gamma(\Delta).V \vdash \phi$. Now, in such a resource proof all the expressions in $\Gamma(\Delta)$ must be mapped to 1 under the corresponding Boolean assignment, and $(\Gamma_1, \Gamma_2)[e] = \Gamma_1[e], \Gamma_2[e]$, so it is clear that there is a resource proof of $\Gamma(\Delta;\Delta').V \vdash \phi$, for which Φ is the corresponding **LBI** proof.

$-\!\!*$L: In this case, the conclusion is $\Delta(\Delta', \Gamma, \phi_1 -\!\!* \phi_2) \vdash \phi$ and the premisses are $\Gamma \vdash \phi_1$ and $\Delta(\Delta', \phi_2) \vdash \phi$, and so by the hypothesis there are disjoint sets of variables V and W such that there are resource proofs of $\Gamma.V \vdash \phi_1$ and $\Delta(\Delta', \phi_2).W \vdash \phi$. Now, as these are resource proofs, by Lemma 4.9, there exists V' such that there are resource proofs of $(\Gamma, \Delta').V' \vdash \phi_1$ and $\Delta((\Gamma, \Delta').\overline{V'}, \phi_2) \vdash \phi$. Hence there is a resource proof R of $\Delta(\Gamma, \Delta', \phi_1 -\!\!* \phi_2) \vdash \phi$ for which Φ is the corresponding **LBI** proof.

$-\!\!*$R: In this case, the conclusion is $\Gamma \vdash \phi_1 -\!\!* \phi_2$ and the premiss is $\Gamma, \phi_1 \vdash \phi_2$ and so, by the hypothesis, there is a resource proof of $(\Gamma, \phi_1[e]).V \vdash \phi_2$, and, as e must be 1, we have that there is a resource proof of $\Gamma.V, \phi_1[1] \vdash \phi_2$; hence there is a resource proof R of $\Gamma.V \vdash \phi_1 -\!\!* \phi_2$, for which Φ is the corresponding **LBI** proof.

$*$L: In this case, the conclusion is $\Gamma(\phi_1 * \phi_2) \vdash \phi$ and the premiss is $\Gamma(\phi_1, \phi_2) \vdash \phi$; so, by the hypothesis, there is a resource proof of $\Gamma(\phi_1, \phi_2).V \vdash \phi$; hence there is a resource proof R of $\Gamma(\phi_1 * \phi_2).V \vdash \phi$, for which Φ is the corresponding **LBI** proof.

$*$R: In this case, the conclusion is $\Gamma, \Delta \vdash \phi_1 * \phi_2$, and the premisses are $\Gamma \vdash \phi_1$ and $\Delta \vdash \phi_2$, and so by the hypothesis there are disjoint sets of variables V and W such that there are resource proofs of $\Gamma.V \vdash \phi_1$ and $\Delta.W \vdash \phi_2$. Now, as these are resource proofs, by Lemma 4.9, there exists V' such that there are resource proofs of $(\Gamma, \Delta).V' \vdash \phi_1$ and $(\Gamma, \Delta).\overline{V'} \vdash \phi_2$. Hence there is a resource proof R of $\Gamma, \Delta \vdash \phi_1 * \phi_2$ for which Φ is the corresponding **LBI** proof.

$\rightarrow$L: In this case, the conclusion is $\Gamma(\phi_1 \rightarrow \phi_2) \vdash \phi$ and the premisses are $\Gamma \vdash \phi_1$ and $\Gamma(\phi_2) \vdash \phi$ and so, by the hypothesis, there are resource proofs of $\Gamma.V \vdash \phi_1$ and $\Gamma(\phi_2[e]).W \vdash \phi$. Now, e must be 1 under W and, as we must have all formulæ in the premisses mapped to 1, we have that there is a set V' of distinct variables V' such that there are resource proofs of $\Gamma.V' \vdash \phi_1$ and $\Gamma(\phi_2[e]).V' \vdash \phi$; hence there is a resource proof R of $\Gamma(\phi_1 \rightarrow \phi_2).V' \vdash \phi$, for which Φ is the corresponding **LBI** proof.

$\rightarrow$R: In this case, the conclusion is $\Gamma \vdash \phi_1 \rightarrow \phi_2$ and the premiss is $\Gamma; \phi_1 \vdash \phi_2$; so, by the hypothesis, there is a resource proof of $(\Gamma; \phi_1[e]).V \vdash \phi_2$ and, as e must be 1, there is a set of distinct variables V' such that there is a resource proof of $\Gamma.V'; \phi_1 \vdash \phi_2$; hence there is a resource proof R of $\Gamma.V' \vdash \phi_1 \rightarrow \phi_2$, for which Φ is the corresponding **LBI** proof.

$\wedge$R: In this case, the conclusion is $\Gamma \vdash \phi_1 \wedge \phi_2$ and the premisses are $\Gamma \vdash \phi_1$ and $\Gamma \vdash \phi_2$; so, by the hypothesis, there are resource proofs of $\Gamma.V \vdash \phi_1$ and $\Gamma.W \vdash \phi_2$. Now, we must have all formulæ in the premisses mapped to 1, and so we have that there is a set V' of distinct variables V' such that there are resource proofs of $\Gamma.V' \vdash \phi_1$ and $\Gamma.V' \vdash \phi_2$; hence there is a resource proof R of $\Gamma.V' \vdash \phi_1 \wedge \phi_2$ for which Φ is the corresponding **LBI** proof.

□

5. STRATEGIES

Having established that the use of Boolean constraints is sound and complete for **MLL** and two extensions of it, we now turn to the issue of how the constraints generated may be solved. As mentioned above, this is independent of the inference rules themselves.

Clearly there are many different search strategies which could be used to generate a solution to the constraints but the strategies which we wish to consider here divides conveniently (but not exhaustively) into three: *lazy*, *eager* and *intermediate*. As mentioned above, resource proofs are intended to be independent of a particular strategy but to contain an explicit specification of the distributive constraints. Strategies are thus distinguished by the manner of solution of the constraints generated during proof-search. We denote as an *n-strategy* one which solves the equations from at most n multiplicative branches at a time.

Lazy distribution. In terms of the calculus introduced above, lazy distribution solves one multiplicative branch's worth of Boolean constraints at a time (thus making it a 1-strategy) and propagates the solution together with any remaining constraints to the next multiplicative branch. This may be thought of as a pessimistic strategy, in that as only a minimal set of constraints is solved, if the derivation turns out to be unsuccessful, then a minimal amount of work has been done. This strategy is the one most commonly used in linear logic programming languages, such as Lygon [Harland et al. 1996; Winikoff and Harland 1995] and Lolli [Hodas and Miller 1994], and is analogous to depth-first search.

Eager distribution. The eager distribution is an ω-strategy, in that an unbounded number of equations may be solved, and so all leaves must be closed before any attempt is made to solve the set of constraints. Hence a constraint solver would be called only once per derivation, with a potentially large set of constraints. This may be thought of as an optimistic strategy, in that if one of the branches leads to failure, then the work done on evaluating all the other branches in parallel has been wasted.[9]

Note that, analogously to the differences between depth-first and breadth-first search, there are examples in which an eager strategy is preferable to a lazy one. For example, consider the sequent $\Gamma \vdash p \otimes q$ where $\Gamma \vdash p$ is provable and q does not occur anywhere in Γ (and hence $\Gamma' \not\vdash q$ for any submultiset Γ' of Γ). Clearly, by an appropriate choice of Γ and p, the proof of $\Gamma \vdash p$ may be made arbitrarily complex (or, for that matter, infinite if $\Gamma \vdash p$ belongs to an undecidable class of sequents). A lazy strategy will generate the sequents

$$\frac{\Gamma \vdash p \quad \vdash q}{\Gamma \vdash p \otimes q}$$

and hence spend an arbitrarily large amount of time on the proof of $\Gamma \vdash p$ (which it will eventually discover is provable) when it is clear that the other branch will fail immediately. However, an eager distribution, which attempts to solve all branches in parallel, will detect that there is no way to close the right-hand branch before any significant amount of work is done, as there is no way to form a putative axiom out of this branch (as q does not appear anywhere in Γ). Hence an eager strategy will be more efficient than a lazy version in this case.

Intermediate distribution. Intermediate strategies are n-strategies, where $n \geq 2$, and are analogous to the technique of *iterative deepening*. The precise way in which a proof which involves $n+1$ multiplicative branches may be either to perform an eager search for the first n such branches (proceeding from the root), and then lazy searches from then on (effectively performing n lazy searches in parallel), or to "switch" the eager version to a place further from the root (effectively performing a number of lazy searches, one of which is a n-way eager search).

In general, it is not possible to predict in advance where the leaves in a proof will be found, so it would seem intuitively reasonable to adopt the policy that the eager behaviour occurs towards the root, and once the bound of n is reached, n multiplicative branches are chosen to be explored in a lazy manner. However, this may result in sub-optimal behaviour, as the "locality" of the constraints is lost. The alternative would require extra analysis, as initially search would proceed as above,

[9]It may also be useful to check that the current constraints have a solution (as distinct from actually solving them), as happens in many constraint logic programming languages.

but once the limit be reached, it would be necessary to re-assign the n searchers to work on the sub-branches of a particular multiplicative branch in some appropriate way.

For example, consider a 2-strategy with the sequent $p, p, q, q \vdash (p \otimes q) \otimes (p \otimes q)$. It is easy to see that as there are 3 occurrences of $\otimes$ in the formula in the succedent, there will be (at least) 4 multiplicative branches in the ensuing proof. Hence it would be reasonable to use the lazy manner for the first occurrence of $\otimes$ and then solve each generated branch in an eager manner.

Other strategies. The three possibilities of lazy, eager and intermediate are probably the most natural choices of strategies but they are not the only ones. One such strategy (which may be thought of as a variation of a purely lazy strategy) may be described as "fact-first", in that given a sequent such as $\Gamma \vdash p \otimes q$, we can note that if $q \in \Gamma$ then one of the multiplicative branches will form an axiom. Hence, rather than arbitrarily selecting one of the branches (the purely lazy strategy), we select the putative axiom, generate the appropriate constraints, and continue with the other branch. Thus the strategy is adaptive, in that the order of evaluation will depend on the sequents generated.

Examples. Consider the **MLL** sequent $p, p, q, q \vdash (p \otimes q) \otimes (p \otimes q)$. A resource proof of this sequent is of the form

$$\dfrac{\dfrac{P_1 \qquad\qquad P_2}{p[x_1], p[x_2], q[x_3], q[x_4] \vdash p \otimes q}\ \otimes R \qquad \dfrac{P_3 \qquad\qquad P_4}{p[\overline{x_1}], p[\overline{x_2}], q[\overline{x_3}], q[\overline{x_4}] \vdash p \otimes q}\ \otimes R}{p, p, q, q \vdash (p \otimes q) \otimes (p \otimes q)}\ \otimes R,$$

where the leaves are as follows:

$P_1 : p[x_1.y_1], p[x_2.y_2], q[x_3.y_3], q[x_4.y_4] \vdash p$

$P_2 : p[x_1.\overline{y_1}], p[x_2.\overline{y_2}], q[x_3.\overline{y_3}], q[x_4.\overline{y_4}] \vdash q$

$P_3 : p[\overline{x_1}.z_1], p[\overline{x_2}.z_2], q[\overline{x_3}.z_3], q[\overline{x_4}.z_4] \vdash p$

$P_4 : p[\overline{x_1}.\overline{z_1}], p[\overline{x_2}.\overline{z_2}], q[\overline{x_3}.\overline{z_3}], q[\overline{x_4}.\overline{z_4}] \vdash q$

The lazy strategy yields the following sequence of constraints and solutions:

Leaf	Constraints added	Solutions
P_1	$x_1.y_1 = 1, x_2.y_2 = 0, x_3.y_3 = 0, x_4.y_4 = 0$	$x_1 = 1, y_1 = 1$
P_2	$x_2.\overline{y_2} = 0, x_3.\overline{y_3} = 1, x_4.\overline{y_4} = 0$	$x_3 = 1, y_3 = 0, x_2 = 0, x_4 = 0$
P_3	$z_2 = 1, z_4 = 0$	$z_2 = 1, z_4 = 0$
P_4	$\overline{x_4}.\overline{z_4} = 1$	

which gives us the overall solution

$$\mathbf{x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0,}$$
$$\mathbf{y_1 = 1, y_2 = 0, y_3 = 0, y_4 = 0,}$$
$$\mathbf{z_1 = 0, z_2 = 1, z_3 = 0, z_4 = 0}$$

where y_2, y_4, z_1 and z_3 have been arbitrarily assigned the value 0.[10] Note that we can conclude from the equations $x_4.y_4 = 0$ and $x_4.\overline{y_4} = 0$ that x_4 must be 0; similarly for x_2.

[10] This is a reflection of the fact that there is more than one solution to the equations.

The eager strategy collects the entire set of equations below and then solves it to produce the same overall solution.

$$\begin{array}{c}
\mathbf{x_1.y_1 = 1, x_2.y_2 = 0, x_3.y_3 = 0, x_4.y_4 = 0}\\
\mathbf{\overline{x_1}.z_1 = 0, \overline{x_2}.z_2 = 1, \overline{x_3}.z_3 = 0, \overline{x_4}.z_4 = 0}\\
\mathbf{x_1.\overline{y_1} = 0, x_2.\overline{y_2} = 0, x_3.\overline{y_3} = 1, x_4.\overline{y_4} = 0}\\
\mathbf{\overline{x_1}.\overline{z_1} = 0, \overline{x_2}.\overline{z_2} = 0, \overline{x_3}.\overline{z_3} = 0, \overline{x_4}.\overline{z_4} = 1}
\end{array}$$

One variant of the intermediate strategy first solves the equations for P_1 and P_3 in parallel and then those for P_2 and P_4:

Leaf	**Constraints added**	**Solutions**
P_1, P_3	$x_1.y_1 = 1, x_2.y_2 = 0, x_3.y_3 = 0, x_4.y_4 = 0$	$x_1 = 1, y_1 = 1$
	$\overline{x_1}.z_1 = 0, \overline{x_2}.z_2 = 1, \overline{x_3}.z_3 = 0, \overline{x_4}.z_4 = 0$	$x_2 = 0, z_2 = 1$
P_2, P_4	$x_2.\overline{y_2} = 0, x_3.\overline{y_3} = 1, x_4.\overline{y_4} = 0$	$x_3 = 1, y_3 = 0$
	$\overline{x_3}.\overline{z_3} = 0, \overline{x_4}.\overline{z_4} = 1$	$x_4 = 0, z_4 = 0$

The other variant of the intermediate strategy first solves the equations for P_1 and P_2 in parallel and then those for P_3 and P_4:

Leaf	**Constraints added**	**Solutions**
P_1, P_2	$x_1.y_1 = 1, x_2.y_2 = 0, x_3.y_3 = 0, x_4.y_4 = 0$	$x_1 = 1, y_1 = 1$
	$x_1.\overline{y_1} = 0, x_2.\overline{y_2} = 0, x_3.\overline{y_3} = 1, x_4.\overline{y_4} = 0$	$x_3 = 1, y_3 = 0, x_4 = 0, x_2 = 1$
P_3, P_4	$z_2 = 1, z_4 = 0$	$z_2 = 1, z_4 = 0$
	$z_2 = 1, \overline{z_4} = 1$	

Note that solving the equations for P_1 and P_2 in parallel generates more of the solution than solving those for P_1 and P_3 in parallel.

Consider now the **PLL** sequent $p, q, q \vdash (p \otimes q) \oplus (p \otimes q \otimes q)$. This has the following resource derivation.

$$\cfrac{\cfrac{\cfrac{P_1 \qquad P_2}{p[y_1], q[y_2], q[y_3] \vdash (p \otimes q)[0], p \otimes q}\otimes\mathrm{R} \qquad p[\overline{y_1}], q[\overline{y_2}], q[\overline{y_3}] \vdash (p \otimes q)[0], q}{p, q, q \vdash (p \otimes q)[x], (p \otimes q \otimes q)[\overline{x}]}\otimes\mathrm{R}}{p, q, q \vdash (p \otimes q) \oplus (p \otimes q \otimes q)}\oplus\mathrm{R}$$

where P_1 and P_2 are as follows:

$$\begin{array}{l}
P_1 : p[y_1.z_1], q[y_2.z_2], q[y_3.z_3] \vdash (p \otimes q)[0], p\\
P_2 : p[y_1.\overline{z_1}], q[y_2.\overline{z_2}], q[y_3.\overline{z_3}] \vdash (p \otimes q)[0], q
\end{array}$$

Solving the equations gives us

$$x = 0, y_1 = 1, y_2 = 1, y_3 = 0, z_1 = 1, z_2 = 0, z_3 = 0$$

Note that $x = 0$ because of the choice of principal formula in the application of the $\otimes$R rule. This gives us the resource proof

$$\cfrac{\cfrac{\cfrac{p[1], q[0], q[0] \vdash (p \otimes q)[0], p \qquad p[0], q[1], q[0] \vdash (p \otimes q)[0], q}{p[1], q[1], q[0] \vdash (p \otimes q)[0], p \otimes q}\otimes\mathrm{R} \qquad p[0], q[0], q[1] \vdash (p \otimes q)[0], q}{p, q, q \vdash (p \otimes q)[0], (p \otimes q \otimes q)[1]}\otimes\mathrm{R}}{p, q, q \vdash (p \otimes q) \oplus (p \otimes q \otimes q)}\oplus\mathrm{R}$$

and hence the sequent proof

$$\dfrac{\dfrac{\dfrac{p \vdash p \quad q \vdash q}{p, q \vdash p \otimes q}\ \otimes\mathrm{R}y \qquad q \vdash q}{p, q, q \vdash p \otimes q \otimes q}\ \otimes\mathrm{R}}{p, q, q \vdash (p \otimes q) \oplus (p \otimes q \otimes q)}\ \oplus\mathrm{R}$$

Note that, in this case, the various strategies will differ just on the sub-proof $p, q, q \vdash p \otimes q \otimes q$ in the manner discussed above.

Turning to an example of an unprovable sequent, consider

$$p \otimes q, r \vdash p \otimes q$$

The only possible resource derivation will look like

$$\dfrac{\dfrac{p[x_1], q[x_2], r[x_3] \vdash p \quad p[\overline{x_1}], q[\overline{x_2}], r[\overline{x_3}] \vdash q}{p, q, r \vdash p \otimes q}}{p \otimes q, r \vdash p \otimes q}\ \otimes\mathrm{L}$$

The lazy strategy would solve, say, the left-hand leaf first, resulting in the equations

$$x_1 = 1, x_2 = 0, x_3 = 0$$

which would make the second leaf into

$$p[0], q[1], r[1] \vdash q$$

which clearly cannot be made into an axiom; the search fails at this point.

The eager strategy would generate the equations

$$x_1 = 1, x_2 = 0, x_3 = 0$$

from the left-hand leaf, together with the equations

$$x_1 = 1, x_2 = 0, x_3 = 1$$

from the second leaf, and hence determine that the union of these two sets of equations has no solution.

Turning to an example from **BI**, consider the sequent

$$(r, (p; t), p \mathrel{-\!\!*} q); s \vdash q * r$$

This results in the following resource derivation:

$$\dfrac{\dfrac{r[x_1], p[x_2] \vdash p}{r[x_1], (p;t)[x_2] \vdash p}\ W \qquad \dfrac{\dfrac{r[\overline{x_1}.y_1], (p;t)[\overline{x_2}.y_2], q[y_3] \vdash q \quad r[\overline{x_1}.\overline{y_1}], (p;t)[\overline{x_2}.\overline{y_2}], q[\overline{y_3}] \vdash r}{r[\overline{x_1}], (p;t)[\overline{x_2}], q \vdash q * r}\ *\mathrm{R}}{(r[\overline{x_1}], (p;t)[\overline{x_2}], q); s \vdash q * r}\ W}{(r, (p;t), p \mathrel{-\!\!*} q); s \vdash q * r}\ \mathrel{-\!\!*}\mathrm{L}$$

Note that the fact that $(r,(p;t),p \mathrel{-\!\!*} q);s$ has an outermost additive context means that the distribution problem for the $\mathrel{-\!\!*}$L rule is to determine how to distribute the multiplicative sub-context $r,(p;t)$.

A lazy evaluation using the leftmost leaf first will give us

$$x_1 = 0, x_2 = 1$$

Propagating this to the next leaf will give us

$$r[y_1],(p;t)[0],q[y_3] \vdash q$$

from which we determine $y_1 = 0, y_3 = 1$. Passing this onto the third leaf gives us

$$r[1],(p;t)[0],q[0] \vdash r$$

as required.

An eager evaluation would generate the set of equations below.

$$x_1 = 0, x_2 = 1, \overline{x_1}.y_1 = 0, \overline{x_2}.y_2 = 0, y_3 = 1, \overline{x_1}.\overline{y_1} = 1, \overline{x_2}.\overline{y_2} = 0, \overline{y_3} = 0$$

Solving this set of equations gives

$$x_1 = 0, x_2 = 1, y_1 = 0, y_2 =, y_3 = 1,$$

as above.

6. THE GENERALITY OF THE METHOD

We have seen that the method of calculating the distribution of formulæ during proof-search with multiplicative rules via solving systems of Boolean constraint equations may be applied to both linear logic and **BI**. Thus these two systems illustrate the use of our methods for the two systems' different treatments of Weakening and Contraction and for the relationship between linear and intuitionistic implication, based both on exponentials and on bunching.

Whilst **BI** can be seen as the free combination of its linear and intuitionistic fragments, the family of "traditional" relevant logics, as represented in [Read 1988; Restall 2000], make choices of connectives and their associated laws which, at least from a semantic perspective, are very much more *ad hoc*.

However, Read's taxonomy [Read 1988] provides a setting for a systematic assessment of the applicability of our methods. Read identifies a basic substructural system, **DW**, and then considers a zoo of possible extensions. We conjecture that our method is sufficiently general to deal with essentially any system involving multiplicative rules. Below, we briefly consider the systematic classification of a large family of systems of relevant logic presented by Read [Read 1988].

The basic system, **DW**, is formulated as a bunched natural deduction system [Read 1988; Pym 2002]. The collection of connectives is essentially the same as that taken in **BI** and the natural deduction system is, of course, closely related to **LBI**. The introduction rules are the same as the right rules of **LBI**. The elimination rules stand in the usual relationship to the left rules of **LBI**. For example, the

Table I. Relevant Axioms

B	$\Gamma,(\Delta,\Theta) \leq (\Gamma,\Delta),\Theta$	prefixing
B′	$\Gamma,(\Delta,\Theta) \leq (\Delta,\Gamma),\Theta$	suffixing
C**	$\Gamma,\emptyset_m \leq \Gamma$	right-identity
C*	$\Gamma,\Delta \leq \Delta,\Gamma$	assertion
C	$(\Gamma,\Delta),\Theta \leq (\Gamma,\Theta),\Delta$	permutation
W*	$\Gamma,\Gamma \leq \Gamma$	conjunctive assertion
W	$(\Gamma,\Delta),\Delta \leq \Gamma,\Delta$	contraction
WB	$\Gamma,(\Gamma,\Delta) \leq \Gamma,\Delta$	conjunctive syllogism
S*	$\Delta,(\Gamma,\Delta) \leq \Gamma,\Delta$	
S	$(\Gamma,\Theta),(\Delta,\Theta) \leq (\Gamma,\Delta),\Theta$	
K*	$\Gamma \leq \Gamma,\Gamma$	mingle / premiss repetition
K	$\Gamma \leq \Gamma,\Delta$	affinity

elimination rules for the multiplicative (intensional) conjunction are implication are, respectively,[11]

$$\otimes\mathrm{E}\quad \frac{\Gamma \vdash \phi \otimes \psi \quad \Delta(\phi,\psi) \vdash \chi}{\Delta(\Gamma) \vdash \chi} \quad \text{and} \quad \multimap \mathrm{E}\quad \frac{\Gamma \vdash \phi \multimap \psi \quad \Delta \vdash \phi}{\Gamma,\Delta \vdash \psi}.$$

The exceptions are the rules for (a classical) negation. The rules for negation are, in our notation,[12]

$$\sim \mathrm{I}\quad \frac{\Gamma,\phi \vdash \sim \psi \quad \Delta \vdash \phi}{\Gamma,\Delta \vdash \sim \phi} \qquad \text{and} \qquad \sim \mathrm{E}\quad \frac{\Gamma \vdash \sim\sim \phi}{\Gamma \vdash \phi},$$

with the optional additional rule,

$$\mathrm{CM}\quad \frac{\Gamma,\phi \vdash \sim \phi}{\Gamma \vdash \sim \phi} \quad \text{“consequentia mirabilis”}.$$

The structural rules of **DW** are:

—The monoid (identity, commutativity, associativity) laws for additive bunching;

—Weakening and Contraction for additive bunching;

—A left-identity for multiplicative bunching, $\emptyset_m,\Gamma \equiv \Gamma$.

Note that the basic system **DW** does not assume commutativity of “,”.

To describe the various additional structural properties which hold within such a system, we write $\Gamma \leq \Gamma'$, for bunches Γ, Γ' to denote that $\Delta(\Gamma) \vdash \phi$ implies $\Delta(\Gamma') \vdash \phi$. The family of relevant logics can then be presented systematically as a hierarchy of systems regulated by the axioms given, using Read's terminology but, as above, our notation, in Table I.

We can now define, following [Read 1988], the following systems:

[11] We use a natural deduction system here in order to conveniently characterize a collection of relevant systems. However, the failure of the subformula property for rules such as $\multimap$ E makes effective search problematic.

[12] We use “,” for multiplicative bunching and “;” for additive bunching; Read's notation is the exact opposite. We write $\emptyset_m$ for the multiplicative identity; Read uses t.

$$
\begin{array}{lll}
\mathbf{TW} = \mathrm{B} + \mathrm{B}' & & \mathbf{DL} = \mathbf{DW} + \mathrm{CM} + \mathrm{WB} \\
\mathbf{RW} = \mathbf{TW} + \mathrm{C} & & \mathbf{TL} = \mathbf{TW} + \mathrm{CM} + \mathrm{WB}\ (= \mathbf{DL} + \mathrm{B} + \mathrm{B}') \\
\mathbf{T} = \mathbf{TL} + \mathrm{W} & & \mathbf{E} = \mathbf{T} + \mathrm{C}^{**} \\
\mathbf{R} = \mathbf{E} + \mathrm{C}^{*} & (= \mathbf{RW} + \mathrm{W}) & \mathbf{RM} = \mathbf{R} + \mathrm{K}^{*}
\end{array}
$$

BI is the system which takes just identity, associativity and commutativity for "," and which takes an implication corresponding to each of "," and ";".

The system **DW**, and so each of the systems defined above, supports the distributive law for additive conjunction and disjunction:

$$\phi \wedge (\psi \vee \chi) \dashv\vdash (\phi \wedge \psi) \vee (\phi \wedge \chi).$$

In this sense, the family of relevant systems is closer to **BI** than to linear logic. Indeed, linear logic without the exponentials may be seen, relative to this hierarchy, as sitting below even **DW**, the rules for its additives not even permitting distribution. Classical logic arises as $\mathbf{R} + \mathrm{K}$, so that we recover a connection, via the exponentials, with the non-distributive, linear systems.

Turning at last to resource proofs, we can make the following observations:

—The natural deduction-style operational rules of **DW** are, apart from negation, all handled just as for their counterparts in **LBI**, the introduction rules being the same as the right rules and the elimination rules typically being simpler than the corresponding left rules. For example, the resource versions of the $\otimes$E and $\multimap$E rules may be formulated as, respectively,

$$\otimes \mathrm{E} \quad \frac{\Gamma.V \vdash \phi \otimes \psi \quad \Delta(\Gamma.\overline{V}, \phi, \psi) \vdash \chi}{\Delta(\Gamma) \vdash \chi} \quad \text{and} \quad \multimap \mathrm{E} \quad \frac{\Gamma.V \vdash \phi \multimap \psi \quad \Gamma.\overline{V} \vdash \phi}{\Gamma \vdash \psi}.$$

The other operational rules, excluding negation, are treated similarly;

—The $\sim$E rule, being unary with only right-hand activity, is trivial. The resource version of the $\sim$I rule is

$$\sim I \quad \frac{\Gamma.V, \phi[e] \vdash \sim \psi \quad \Gamma.\overline{V} \vdash \phi}{\Gamma \vdash \sim \phi} \quad e = 1.$$

The CM rule goes as

$$\mathrm{CM} \quad \frac{\Gamma, \phi[e] \vdash \sim \phi}{\Gamma \vdash \sim \phi} \quad e = 1;$$

—The various structural rules can now be seen to be unproblematic. For example, W^*, W, WB, S^* and S all follow the pattern of Contraction in **BI**. Similarly, K^* and K follow the pattern of Weakening in **BI**.

Our usual soundness and completeness properties can now be established for the family of relevant systems .

Thus our analysis encompasses the full range of established (propositional) substructural logics. We conjecture that our methods can be applied to the multiplicative (or intensional) quantifiers introduced in [O'Hearn and Pym 1999; Pym 1999] and developed in [Pym 2002]. The usual additive (or extensional) quantifiers were treated in [Harland and Pym 1997] (they are unproblematic).

ACKNOWLEDGMENTS

We thank John Crossley, Samin Ishtiaq, Michael Winikoff and several anonymous referees for their comments on this work. Harland is grateful for the hospitality of the Department of Computer Science of Queen Mary, University of London during a period of sabbatical leave, and for the partial support of the British Council. Pym is grateful for the partial support of the UK EPSRC and of Queen Mary, University of London.

REFERENCES

ANDREOLI, J.-M. 1992. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation 2,* 3.

ANDREOLI, J.-M. 2001. Focussing and proof construction. *Annals of Pure and Applied Logic 107,* 1-3, 131–163.

ARMELÍN, P. AND PYM, D. 2001. Bunched logic programming (extended abstract). In *Proceedings of IJCAR 2001.* Lecture Notes in Artificial Intelligence 2083, 289–304.

CERVESATO, I., HODAS, J. S., AND PFENNING, F. 2000. Efficient resource management for linear logic proof search. *Theoretical Computer Science 232,* 1–2 (Feb.). Special issue on Proof Search in Type-Theoretic Languages, D. Galmiche and D. Pym, editors.

GABBAY, D. 1996. *Labelled Deductive Systems: Principles and Applications, Vol. 1: Basic Principles.* Oxford University Press.

GALMICHE, D. 2000. Connection methods in linear logic and proof nets construction. *Theoretical Computer Science 232*, 231–272.

GALMICHE, D. AND MÉRY, D. 2001a. Proof-search and countermodel generation in propositional **bi** logic (extended abstract). In *Fourth International Symposium on Theoretical Aspects of Computer Software, TACS 2001.* LNCS, Sendai, Japan.

GALMICHE, D. AND MÉRY, D. 2001b. *Semantic Tableaux for* **BI***, I.* Manuscript.

GALMICHE, D., MÉRY, D., AND PYM, D. 2001. *Semantic Tableaux for* **BI***, II.* In preparation.

GIRARD, J.-Y. 1987. Linear logic. *Theoretical Computer Science 50*, 1–102.

GIRARD, J.-Y. 1994. *Proof-nets for Additives.* Manuscript.

HARLAND, J. AND PYM, D. 1997. Resource-distribution via boolean constraints (extended abstract). In *Proceedings of the International Conference on Automated Deduction (CADE-14)*, W. McCune, Ed. Lecture Notes in Computer Science 1249, Townsville, 222–236.

HARLAND, J., PYM, D., AND WINIKOFF, M. 1996. Programming in lygon: An overview. In *Proceedings of the International Conference on Algebraic Methodology and Software Technology (AMAST)*, M. Wirsing and M. Nivat, Eds. Lecture Notes in Computer Science 1101, 391–405.

HODAS, J. AND MILLER, D. 1994. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation 110,* 2, 327–365.

KRIPKE, S. 1965. Semantical analysis of intuitionistic logic. In *Formal Systems and Recursive Functions*, J. Crossley and M. Dummett, Eds. Amsterdam, 92–130.

O'HEARN, P. AND PYM, D. 1999. The logic of bunched implications. *Bulletin of Symbolic Logic 5,* 2 (June), 215–244.

PYM, D. 1999. On bunched predicate logic. In *Proceedings of the 14th IEEE Symposium on Logic in Computer Science.* IEEE Computer Society, Trento, Italy, 183–192.

PYM, D. 2002. *The Semantics and Proof Theory of the Logic of Bunched Implications.* Kluwer Academic Publishers, Applied Logic Series.

PYM, D. AND HARLAND, J. 1994. A uniform proof-theoretic investigation of linear logic programming. *Journal of Logic and Computation 4,* 2 (Apr.), 175–207.

PYM, D., O'HEARN, P., AND YANG, H. 2000. *Possible worlds and resources: the semantics of* **BI**. Manuscript.

READ, S. 1988. *Relevant Logic.* Blackwell.

RESTALL, G. 2000. *An Introduction to Substructural Logics.* Routledge.

TAMMET, T. 1994. Proof search strategies in linear logic. *Journal of Automated Reasoning 12*, 273–304.

TROELSTRA, A. 1992. *Lectures on Linear Logic.* CSLI Lecture Notes No. 29.

URQUHART, A. 1972. Semantics for relevant logics. *Journal of Symbolic Logic 37*, 159–169.

WINIKOFF, M. AND HARLAND, J. 1995. Implementing the linear logic programming language lygon. In *Proceedings of the International Logic Programming Symposium*, J. Lloyd, Ed. MIT Press, Portland, 66–80.

Received December 2000; revised August 2001; accepted October 2001