An Algebraic Approach to Proof Search in Sequent Calculi

James Harland

Department of Computer Science
Royal Melbourne Institute of Technology
GPO Box 2476V
Melbourne 3001, Australia.
Email: jah@cs.rmit.edu.au
WWW: www.cs.rmit.edu.au/∼jah

Abstract. Inference rules in the sequent calculus can be interpreted as both proof construction rules and proof search rules. However, the kind of information used in each case is somewhat different. In this paper we explore these differences by using a multiple-conclusioned sequent calculus for intuitionistic logic (LM) as a search calculus for proofs in the single-conclusioned intuitionistic sequent calculus LJ. We also show how the classical sequent calculus LK can be considered as both a search calculus and a proof calculus. The key technical issue is to determine the appropriate Boolean constraints to be attached to each formula, reflecting the search choices made. We also briefly discuss the possibilities for extending these techniques to intermediate logics via Avron's hypersequents, and to an additive form of proof-nets for linear logic.

1 Introduction

It is well known that the rules of inference in the sequent calculus can be considered as both rules of *deduction* and of *reduction*. For example, consider the $\wedge$R rule below from the classical sequent calculus LK.

$$\frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \wedge \psi, \Delta} \wedge R$$

As a rule of deduction, this states that from a proof of $\Gamma \vdash \phi, \Delta$ and a proof of $\Gamma \vdash \psi, \Delta$ we can construct a proof of $\Gamma \vdash \phi \wedge \psi, \Delta$. As a rule of reduction, this states that in order to find a proof of $\Gamma \vdash \phi \wedge \psi, \Delta$ it is sufficient to find a proof of $\Gamma \vdash \phi, \Delta$ and a proof of $\Gamma \vdash \psi, \Delta$. Hence the same inference rule can be interpreted as a means of leaf-to-root proof construction, or as a means of root-to-leaf proof search.

In this paper we consider in further detail this difference in emphasis between deduction and reduction. In particular we develop a search-oriented variation of the intuitionistic sequent calculus in order to sharpen the distinction between proof search and proof construction. This distinction is based on observations of the differences between the single-conclusioned intuitionistic calculus LJ and the multiple-conclusioned calculus, which, following Ritter, Pym and Wallen [10], we refer to as *LM* [11, 12].

In [6] it is noted how LM can be considered as better for search than LJ, due to the greater possibilities for delaying choices (particularly for disjunctions) in LM than LJ. As in [7] for the resource distribution problem of linear logic, we use Boolean constraints to keep track of certain internal details of the choices made during search, in order to extract a maximally informative proof; we may thus conceive of a partial proof containing such constraints as a *search tree* from which, on the successful completion of the search, we can calculate a *proof tree*. Thus a proof tree can be considered a special case of a search tree.

Specifically, we give a search-oriented version of the inference rules for LM together with a method of extracting an LJ proof from a successful search for an LM proof.

This extraction is not merely a matter of deleted unwanted choices; the "shape" of a proof in LM may be different to that of the LJ proof extracted from it. Hence the extraction may involve rearrangement of the inferences rules used in the LM proof.

We also show how the same techniques can be applied to LK, thus providing a system which can search for LK or LM proofs (and, in fact, for LJ proofs directly if desired), thus generating an instance of *labelled deduction* [2]. We also discuss some possibilities for further work, which include extensions of these techniques to intermediate logics (via Avron's *hypersequents* [1]), and to an additive form of proof-nets for linear logic[4, 3].

2 Proof Search

LM's suitability for proof search stems from its greater number of *local* constraints than LJ. LJ has an overall requirement that there is at most one formula in a succedent; LM merely requires that when one of the rules ∀R, →R and ¬R is used, then the formulae other than the principal one need to be discarded from the succedent.

For example, consider the LJ and LM proofs below of $p, p \to q \vdash q \vee r$.

$$\dfrac{\dfrac{p, q \vdash q \quad p \vdash p}{p, p \to q \vdash q}\to L}{p, p \to q \vdash q \vee r}\vee R \qquad \dfrac{\dfrac{p, q \vdash q, r \quad p \vdash p, r}{p, p \to q \vdash q, r}\to L}{p, p \to q \vdash q \vee r}\vee R$$

Note the "magic" choice of q in the LJ ∨R step; in general, this reflects "don't know" nondeterminism, in that a search procedure will need to backtrack over this choice in the event of a subsequent failure. In the LM proof, there is no "magic" choice made in the ∨R rule; both possibilities are available to the next step, and hence the choice between the two formulae is delayed until further information is available (in this case, at the leaves of the proof).

In order to recover an LJ proof from an LM proof, we will attach a Boolean expression, as in [7], to each formula in the proof, and use the information thus recorded to keep track of the status of each formula, and in particular which choice of formula has been made. The values of the Boolean expressions are typically defined by the leaves of the proof (i.e. the axioms) and the choice of rules made in the construction of the proof. We will refer to such derivation trees as *search trees*.

For example, the search tree version of the above LM proof is

$$\dfrac{\dfrac{p[x_1], q[y_1] \vdash q[z_1], r[z_2] \quad p[x_2] \vdash p[y_2], r[z_2]}{p[x], p \to q[y] \vdash q[z_1], r[z_2]}\to L}{p[x], p \to q[y] \vdash q \vee r[z]}\vee R$$

From the axioms we determine that $x_1 = 0, y_1 = z_1 = 1, z_2 = 0, x_2 = y_2 = 1, z_2 = 0$ and thus we have the "completed" search tree and LJ proof as below.

$$\dfrac{\dfrac{p[0], q[1] \vdash q[1], r[0] \quad p[1] \vdash p[1], r[0]}{p[1], p \to q[1] \vdash q[1], r[0]}\to L}{p[1], p \to q[1] \vdash q \vee r[1]}\vee R \qquad \dfrac{\dfrac{q \vdash q \quad p \vdash p}{p, p \to q \vdash q}\to L}{p, p \to q \vdash q \vee r}\vee R$$

Clearly we can obtain the LJ proof by deleting formulae marked with 0, as these have been determined to be not necessary in the proof. Note that this proof is slightly more precise than the original one above; in the latter LJ proof it is been detected that p is not required in the left hand branch of →L. In this manner we may think of LM as a *search calculus* for intuitionistic logic, and LJ as an *object calculus*, in that for each proof in the search calculus, we can find a corresponding proof in the object calculus (and vice-versa).

Clearly it will be straightforward to generate an LJ proof from the search tree if there is at most one formulae in each succedent of the proof which is assigned the value 1. If this is not the case, then rearrangement of the proof will generally be necessary in order to find an LJ proof. For example, consider the LM proof P_1 below and an LJ proof of the same sequent.

$$
\dfrac{\dfrac{\dfrac{\dfrac{r,p \vdash p,q,s \quad r,q \vdash p,q,s}{r,p \vee q \vdash p,q,s}\,\vee L}{r \wedge (p \vee q) \vdash p,q,s}\,\wedge L \qquad s \vdash q,s}{r \wedge (p \vee q), p \to s \vdash s,q}\,\to L}{r \wedge (p \vee q), p \to s \vdash s \vee q}\,\vee R
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{r,p \vdash p \quad r,p,s \vdash s}{r,p,p \to s \vdash s}\,\to L}{r,p,p \to s \vdash s \vee q}\,\vee R \qquad \dfrac{r,q,p \vdash q}{r,q,p \to s \vdash q}\,\vee R}{r,p \vee q, p \to s \vdash s \vee q}\,\vee L}{r \wedge (p \vee q), p \to s \vdash s \vee q}\,\wedge L
$$

Note that in the LJ proof, the ∨L instance is closer to the root than both the →L instance and the ∨R instance. We will consider such issues in more detail in the next section.

Note that we introduce a fresh variable for each active formula. For example, in the search tree above for the sequent $p[x], p \to q[y] \vdash q \vee r[z]$, we find that $z = 1$ because it is the principal formula of the ∨R instance, but the active formulae in the premise of this rule instance are $q[z_1], r[z_2]$, reflecting the fact that both, exactly one or neither of these formulae may be essential to any proof which is ultimately found by this search process. This is in contrast to the system of [7], which, owing to its basis in linear logic, requires that active formulae be used exactly once; in LM (and LJ and LK for that matter) this restriction does not apply, and so active formulae are not necessarily used in the proof.

A similar "refreshment" of the variables is required for binary rules (∧R, ∨L, →L). For example, consider the search tree below.

$$
\dfrac{\dfrac{p[w_1], s[z_1] \vdash p[x_{11}], q[x_{12}], r[y_1] \quad p[w_2], s[z_2] \vdash p[x_{21}], q[x_{22}], s[y_2]}{p[w], s[z] \vdash p[x_1], q[x_2], r \wedge s[y]}\,\wedge R}{p[w], s[z] \vdash p \vee q[x], r \wedge s[y]}\,\vee R
$$

Here we note that the active formulae all get fresh variables when they appear (such as $p[x_1], q[x_2]$ from the ∨R instance, $r[y_1]$ and $s[y_2]$ from the ∧R instance), and also that the context formulae get a fresh variable each time a branch is made in the proof, as the behaviour of the formulae on each branch is independent. In this case the branch is due to the ∧R rule and the formulae $p[w], s[z]$ in the antecedent and $p[x_1], q[x_2]$ in the succedent get fresh variables on each of the branches.

By solving the equations it is then straightforward to obtain

$$
\dfrac{\dfrac{p[1], s[0] \vdash p[1], q[0], r[0] \quad p[0], s[1] \vdash p[0], q[0], s[1]}{p[1], s[1] \vdash p[1], q[0], r \wedge s[1]}\,\wedge R}{p[1], s[1] \vdash p \vee q[1], r \wedge s[1]}\,\vee R
$$

Note that as both p and s (in the antecedent) are used on exactly one of the branches of the $\wedge$R instance their value in the conclusion of this rule is 1. However, as q is used on neither branch, it is value in the same sequent is 0. Thus in order to calculate the final value for context formulae in the conclusion of a binary rule, we take the Boolean sum of the values one each branch.

From this completed search tree it is not hard to see that as one of the active formulae of the $\wedge$R is not used in the proof that the application of this rule is unnecessary, and in fact we can simplify the proof of this sequent to the one on the left below, and hence to the LJ proof on the right below.

$$\frac{p[1], s[0] \vdash p[1], q[0]}{p[1], s[0] \vdash p \vee q[1]} \vee R \qquad \frac{p, s \vdash p}{p, s \vdash p \vee q} \vee R$$

Note that the value assigned to s is now 0, as the value was previously 1 due to the deleted axiom. Hence as a general rule, we wish to delete from the proof any rule instances in which the active formulae are assigned the value 0. The only exception to this is the rule $\vee$R, and then only when exactly one of the active formulae is assigned the value 0. For this proof it is clear that this condition holds for the active formulae of $\vee$R (i.e. $p[1], q[0]$) which has a clear correspondence with the LJ version.

However, if both active formulae are assigned 0, then any such instance of $\vee$R is also to be deleted.

The special rules (i.e. $\forall$R, $\rightarrow$R and $\neg$R) are also worthy of note. In general it is not known in advance which choice of rule instance (and hence of principal formula) will lead to a proof, if any. Hence we represent this choice by requiring not only that the principal formula of the selected rule instance be assigned the value 1, but that all other formulae in the succedent be assigned the value 0.

3 Extracting an LJ proof

In order to extract an LJ proof from a search tree, we calculate the maximum number of formulae assigned 1 in any succedent in the proof; our goal is then to reduce this value to 1 by various manipulations on the proof. Note that such manipulations may alter the root of the proof; in other words, a successful search tells us that we have found something, but the extraction process tells us exactly what was found.

The first step is to permute instances of $\vee$L as close to the root of the proof as possible. As $\vee$L is *invertible*, this can always be done, except when the principal formula is an active formula of a subsequent rule instance.

Lemma 1. *If $\Gamma, F_1 \vee F_2 \vdash \Delta$ has an LM-proof, then there is an LM proof in which $\vee$L is at the root.*

Hence we have that any instance of $\vee$L in the proof after this transformation must either be at the root of the proof or separated from it only by other instances of $\vee$L, or be immediately above an inference in which the principal formula of $\vee$L is an active formula of the subsequent rule. For example, in the LM proof P_1 above, $p \vee q$ is an active formulae of the instance of $\wedge$L immediately below it.

Now any such rule will necessarily not have an active formula in the succedent of the conclusion of the $\lor$L rule (as there is no rule which has active formulae in both the antecedent and succedent *of the same sequent*); consequently any right-active rule immediately below this sequence of left-active rules with $\lor$L uppermost can be permuted upwards past this occurrence of $\lor$L.

Note also that the only rules which have a left-active formula but with the principal formula in the succedent are $\to$R and $\neg$R; for such rules to be applicable, there can be at most one formula in the succedent, and hence there is no need to rearrange the proof.

For example, consider the LM proof below and a permutation of it in which the $\neg$L instance is pushed upwards.

$$\dfrac{\dfrac{\dfrac{\dfrac{r,p \vdash p,q,s \quad r,q \vdash p,q,s}{r,p \lor q \vdash p,q,s}\,\lor L}{r \land (p \lor q) \vdash p,q,s}\,\land L \qquad s \vdash q,s}{r \land (p \lor q), p \to s \vdash s,q}\,\to L}{r \land (p \lor q), p \to s, \neg q \vdash s}\,\neg L \qquad \dfrac{\dfrac{\dfrac{r,p,\neg q \vdash p,s \quad \dfrac{r,q \vdash p,q,s}{r,\neg q,q \vdash p,s}\,\neg L}{r,p \lor q,\neg q \vdash p,s}}{r \land (p \lor q), \neg q \vdash p,s}\,\land L \qquad s \vdash s}{r \land (p \lor q), p \to s, \neg q \vdash s}\,\to L$$

However, pushing the $\neg$L upwards in this way makes the instance of $\to$L more "LJ-like" (and similarly for the other binary rules $\lor$L and $\land$R). In the absence of $\lor$L, it is straightforward to show that whenever $\Gamma \vdash_{LM} \Delta$, $\exists F \in \Delta$ such that $\Gamma \vdash_{LM} F$. As this property also holds at the leaves of an arbitrary LM proof, the only way to get more than one "necessary" formula in a succedent is via an instance of $\lor$L. Once this state is achieved, any of the binary rules $\to$L $\land$R and $\lor$L could increase this number.

However, if we can always permute the instances of $\to$L and $\land$R above the instance of $\lor$L, any increase will be solely due to $\lor$L, in which case we can immediately precede such instances of $\lor$L by an appropriate pair of instances of $\lor$R.

For example given the proof on the left below, we can permute the $\to$L instance above the $\land$L instance and the $\lor$L instance to get the proof on the right below.

$$\dfrac{\dfrac{\dfrac{r,p \vdash p,q,s \quad r,q \vdash p,q,s}{r,p \lor q \vdash p,q,s}}{r \land (p \lor q) \vdash p,q,s}\,\land L \qquad s \vdash q,s}{r \land (p \lor q), p \to s \vdash s,q}\,\to L \qquad \dfrac{\dfrac{\dfrac{r,p \vdash p \quad r,p,s \vdash s}{r,p,p \to s \vdash s}\,\to L \qquad r,q,p \to s \vdash s}{r,p \lor q, p \to s \vdash q,s}\,\lor L}{r \land (p \lor q), p \to s \vdash s,q}\,\land L$$

Then it is simple enough to insert two $\lor$R instances to obtain the LJ proof below.

$$\dfrac{\dfrac{\dfrac{\dfrac{r,p \vdash p \quad r,p,s \vdash s}{r,p,p \to s \vdash s}\,\to L}{r,p,p \to s \vdash s \lor q}\,\lor R \qquad \dfrac{r,q,p \to s \vdash q}{r,q,p \to s \vdash s \lor q}\,\lor R}{r,p \lor q, p \to s \vdash s \lor q}\,\lor L}{r \land (p \lor q), p \to s \vdash s \lor q}\,\land L$$

Hence the extraction of the LJ proof is as follows.

1. Permute $\lor$L inferences as close to the root as possible
2. Permute right-active rules as close to the leaves as possible

3. Delete formulae marked 0
4. Delete any rule inferences in which at least one active formula is marked 0 (except for $\vee$R, in which case only delete those in which both active formulae are marked 0)
5. Insert instances of $\vee$R immediately above any instances of $\vee$L.

4 Formal Results

Below we give the formal results which established that the above extraction process will always result in an LJ proof.

Definition 1. *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. We denote by $\Gamma[X]$ where Γ is an unannotated set of formulae the set $\{\phi[x] \mid \phi \in \Gamma, x \in X\}$. A sequent consisting entirely of annotated formulæ is known as a* search sequent.

Definition 2. *Given a multiset of annotated formulæ $\Delta = \{\phi_1[e_1], \dots \phi_n[e_n]\}$ and a total assignment I of the Boolean variables in Δ, we define $\Delta[I] = \{\phi_1[v_1], \dots \phi_n[v_n]\}$, 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 $\phi[1]$; it will always be possible to disambiguate such annotations from the context.

Definition 3.

We say a binary rule R obeys the binary rule search condition *if the context formulae $\Gamma \cup \Delta$ are such that for each $F \in \Gamma \cup \Delta$ in the conclusion of the rule, we have that exp$(F) = x$ in the conclusion and $x = x_1 + x_2$, where x_1 = exp(F) in the left premise and x_2 = exp(F) in the right premise.*

We now turn to the specification of the search rules for LM.

Definition 4. *An* LM-resource derivation *is built via the following rules:*

$$\frac{}{\Gamma, F[x] \vdash F[y], \Delta}\text{ Axiom} \qquad x = y = 1, \forall z \ \in \exp(\Gamma \cup \Delta)\ z = 0$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, F[z] \vdash \Delta}\text{ WL} \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash F[z], \Delta}\text{ WR} \quad \frac{\Gamma, F[x], F[y] \vdash \Delta}{\Gamma, F[z] \vdash \Delta}\text{ CL} \quad \frac{\Gamma \vdash F[x], F[y], \Delta}{\Gamma \vdash F[z], \Delta}\text{ CR}$$

$$\frac{\Gamma, F_1[x], F_2[y] \vdash \Delta}{\Gamma, F_1 \wedge F_2[z] \vdash \Delta}\wedge\text{L} \qquad \frac{\Gamma[X'] \vdash F_1[x], \Delta[Y'] \quad \Gamma[X''] \vdash F_2[y], \Delta[Y'']}{\Gamma[X] \vdash F_1 \wedge F_2[z], \Delta[Y]}\wedge\text{R}$$

$$\frac{\Gamma[X'], F_1[x] \vdash \Delta[Y'] \quad \Gamma[X''], F_2[y] \vdash \Delta[Y'']}{\Gamma[X], F_1 \vee F_2[z] \vdash \Delta[Y]}\vee\text{L} \qquad \frac{\Gamma \vdash F_1[x], F_2[y], \Delta}{\Gamma \vdash F_1 \vee F_2[z], \Delta}\vee\text{R}$$

$$\frac{\Gamma[X'] \vdash F_1[x], \Delta[Y'] \quad \Gamma[X''], F_2[y] \vdash \Delta[Y'']}{\Gamma[X], F_1 \to F_2[z] \vdash \Delta[Y]} \to\text{L} \qquad \frac{\Gamma, F_1[x] \vdash F_2[y], \Delta}{\Gamma \vdash F_1 \to F_2[z], \Delta} \to\text{R}$$

$$\frac{\Gamma, F[y/x][w] \vdash \Delta}{\Gamma, \exists x F[z] \vdash \Delta}\ \exists\text{L} \quad \frac{\Gamma \vdash F[t/x][w], \Delta}{\Gamma \vdash \exists x F[z], \Delta}\ \exists\text{R} \quad \frac{\Gamma, F[t/x][w] \vdash \Delta}{\Gamma, \forall x F[z] \vdash \Delta}\ \forall\text{L} \quad \frac{\Gamma \vdash F[y/x][w], \Delta}{\Gamma \vdash \forall x F[z], \Delta}\ \forall\text{R}$$

$$\frac{\Gamma \vdash F[x], \Delta}{\Gamma, \neg F[z] \vdash \Delta}\ \neg\text{L} \quad \frac{\Gamma, F[x] \vdash \Delta}{\Gamma \vdash \neg F[z], \Delta}\ \neg\text{R}$$

where

- $z = 1$ *for each rule other than Axiom*
- *the rules* $\exists$*L and* $\forall$*R have the side condition that* y *is not free in* Γ, Δ *or* F
- *the rules* $\vee$*L,* $\rightarrow$*L and* $\wedge$*R obey the binary rule search condition and* X', X'', Y' *and* Y'' *are all distinct sets of Boolean variables*
- *in the succedent of the premise of the rules* $\forall$*R,* $\rightarrow$*R and* $\neg$*R we have that* $\forall x \in \exp(\Delta)\ x = 0$

Definition 5. *A* search tree *is a tree regulated by the rules of the search 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 tree.*

A search tree is total *if its assignment of the Boolean variables is total. Otherwise, the search tree is* partial*. A search tree is* closed *if all of the leaves of the tree are instances of one of the leaf rules. A* proof tree *is a total, closed search tree.*

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

It is not hard to show the following result. The proof is straightforward and hence omitted.

Proposition 1. $\Gamma \vdash \Delta$ *has a proof in LM iff there is a proof tree of a corresponding search sequent.*

As mentioned above, the aim is to produce an LM proof which is single-conclusioned and hence an LJ proof. We will use the *multiplicity* of the proof as a measure of how far we are from an LJ proof.

Definition 6. *We denote by the* multiplicity *of a LM resource-proof the maximum number of formulae in a succedent in the resource-proof whose value is 1.*

Hence we come to the main result.

Proposition 2. *If* $\Gamma \vdash \Delta$ *has an LM proof tree with Boolean assignment* I*, then there is an LJ proof of* $\Gamma \vdash \bigvee(\Delta')$*, where* $\Delta' \subseteq \Delta[I]^1$.

Proof. (Sketch).

Our aim is to generate a proof tree of multiplicity 1.

After steps 1 and 2 of the above conversion process, when travelling from leaf to root, the multiplicity of the proof can only be increased by an instance of $\vee$L. At such rule instances, the discussion above shows that any subsequent right-active rules (including $\rightarrow$L and $\wedge$R) are either inapplicable or can be permuted upwards, and so any formula in the succedent of the conclusion of $\vee$L is not an active formula of any subsequent rule in the proof. Hence all that remains is to delete formulae marked 0 (step 3) and any rules which are now inapplicable (step 4) before collecting the formulae of the succedent of the conclusion of $\vee$L into one disjunction (step 5), thus generating a proof of multiplicity 1.

5 Adding Classical Proof Search

It is not hard to extend the above search calculus to one for LK; after all, one of the main features of LM is that the only differences between LM and LK is in the three rules $\forall$R, $\rightarrow$R and $\neg$R. LK may thus be considered as both a search calculus and a proof calculus. In addition, given such a dual system, it is again not difficult to extend it to include a direct means for the recognition of LJ proofs, i.e. constraints based on LJ rather than on LM as above. We thus arrive at a search calculus for all three systems LK, LM and LJ; below we present a selection of rules for this unified system, together with a table indicating the corresponding Boolean constraints for each rule. Note that the constraints for LM and LJ are in addition to those for LK.

$$\frac{}{\Gamma[X], F[x] \vdash F[z], \Delta[Z]}\ \text{Axiom} \qquad \frac{\Gamma \vdash \Delta[X]}{\Gamma \vdash F[z], \Delta[Z]}\ \text{WR} \qquad \frac{\Gamma \vdash F[x], F[y], \Delta[X]}{\Gamma \vdash F[z], \Delta[Z]}\ \text{CR}$$

$$\frac{\Gamma, F_1[x] \vdash \Delta[X] \quad \Gamma, F_2[y] \vdash \Delta[Y]}{\Gamma, F_1 \vee F_2[z] \vdash \Delta[Z]}\ \vee\text{L} \qquad \frac{\Gamma \vdash F_1[x], F_2[y], \Delta[Y]}{\Gamma \vdash F_1 \vee F_2[z], \Delta[Z]}\ \vee\text{R} \qquad \frac{\Gamma, F[x] \vdash \Delta[X]}{\Gamma \vdash \neg F[z], \Delta[Z]}\ \neg\text{R}$$

$$\frac{\Gamma \vdash F_1[x], \Delta[X] \quad \Gamma, F_2[y] \vdash \Delta[Y]}{\Gamma, F_1 \rightarrow F_2[z] \vdash \Delta[Z]}\ \rightarrow\text{L} \qquad \frac{\Gamma, F_1[x] \vdash F_2[y], \Delta[X]}{\Gamma \vdash F_1 \rightarrow F_2[z], \Delta[Z]}\ \rightarrow\text{R} \qquad \frac{\Gamma \vdash (F[y/x])[w], \Delta[X]}{\Gamma \vdash \forall x F[z], \Delta[Z]}\ \forall\text{R}$$

	Axiom	WR	CR	$\vee$L	$\vee$R	$\neg$R	$\rightarrow$L	$\rightarrow$R	$\forall$ R
LK	$x = z = 1$ $X = Z = 0$	$z = 1$	$z = 1$	$z = 1$	$z = 1$	$z = 1$	$z = 1$	$z = 1$	$z = 1$
LM	-	-	-	-	-	$X = 0$	-	$X = 0$	$X = 0$
LJ	-	$X = 0$ $Z = 0$	$0 = 1$	$X = Y$ $X = Z$	$Z = Y = 0$ $z = 1, x = \overline{y}$	$X = 0$ $Z = 0$	$X = 0$ $Y = Z$	$X = 0$ $Z = 0$	$X = 0$ $Z = 0$

Hence given a classically provable but intuitionistically unprovable sequent such as $\vdash ((p \rightarrow q) \rightarrow p) \rightarrow p$ (Peirce's formula) we get an LK search tree and proof tree as follows.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p[x_7] \vdash q[x_6], p[x_1]}\ \text{Ax}}{\vdash p \rightarrow q[x_3], p[x_1]}\ \rightarrow\text{R} \qquad \dfrac{}{p[x_5] \vdash p[x_4]}\ \text{Ax}}{(p \rightarrow q) \rightarrow p[x_2] \vdash p[x_1]}\ \rightarrow\text{L}}{\vdash ((p \rightarrow q) \rightarrow p) \rightarrow p[x]}\ \rightarrow\text{R} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{p[1] \vdash q[0], p[1]}\ \text{Ax}}{\vdash p \rightarrow q[1], p[1]}\ \rightarrow\text{R} \qquad \dfrac{}{p[1] \vdash p[1]}\ \text{Ax}}{(p \rightarrow q) \rightarrow p[1] \vdash p[1]}\ \rightarrow\text{L}}{\vdash ((p \rightarrow q) \rightarrow p) \rightarrow p[1]}\ \rightarrow\text{R}$$

Here it is straightforward to note that this is not an LM proof, as the above instance of $\rightarrow$R does not satisfy the LM constraint for $\rightarrow$. Note that a corresponding LM search attempt will satisfy the $\rightarrow$L constraint for LM, but will not lead to a proof.

6 Conclusions and Further Work

We have shown how Boolean constraints can be used, in the manner of [7] to develop a search calculus for intuitionistic logic based on LM, and from which one can extract LJ proofs. In some ways the extraction result is not all that strong, in that once an LM proof is known to exist, the soundness and completeness of both LM and LJ for intuitionistic logic ensures that some LJ proof exists. The main contribution here is to introduce the notion of a search calculus, in which it is simple enough to extract an LJ proof from a successful search in LM. A more precise specification of the relationship between the extracted LJ proof and the LM proof is an obvious piece of further work, particularly from the point of view of search spaces.

Another interesting direction to pursue is to compare this approach with one based on Avron's *hypersequents* [1]. The latter are generally introduced as an elegant means of providing rules of inference for intermediate logics such as Gödel-Dummett logic. This is done by introducing an inter-sequent operator $|$ which intuitively acts as a disjunction:

$$\Gamma_1 \vdash F_1 \mid \Gamma_2 \vdash F_2$$

is provable iff $\Gamma_i \vdash F_i$ is provable for some $i = 1, 2$.

Hence an investigation of hypersequents as a search calculus seems interesting. In particular, a hypersequential form of the $\forall$R, $\rightarrow$R and $\neg$R rules of LM, along the lines of the $\rightarrow$R rule below, would be of particular interest.

$$\frac{\Gamma, F_1 \vdash F_2 \mid \Gamma \vdash \Delta}{\Gamma \vdash F_1 \rightarrow F_2, \Delta}$$

Hypersequents are also interesting for their relationship to Gödel-Dummett logic. In [5] it is shown how this logic naturally arises from considerations of equivalence in logic programs; given the emphasis on search strategies in LJ (as in [9], amongst others) in the derivation of logic programming languages, a natural conclusion is to perform analyses of logic programming languages in a search calculus for Gödel-Dummett logic. A hypersequential version of LM would be an excellent starting point for such an investigation.

A further point of interest is to extend the original system of [7] to more than just the multiplicative fragment of linear logic. In particular, the additive rules $\&$R and $\oplus$R

$$\frac{\Gamma \vdash F_1, \Delta \quad \Gamma \vdash F_2, \Delta}{\Gamma \vdash F_1 \,\&\, F_2, \Delta}\ \&R \qquad \frac{\Gamma \vdash F_i, \Delta}{\Gamma \vdash F_1 \oplus F_2, \Delta}\ \&R$$

would become

$$\frac{\Gamma \vdash F_1[x], F_2[\overline{x}], \Delta}{\Gamma \vdash F_1 \,\&\, F_2, \Delta}\ \&R \qquad \frac{\Gamma \vdash F_1[y], F_2[\overline{y}], \Delta}{\Gamma \vdash F_1 \oplus F_2, \Delta}\ \&R$$

where x is universally quantified, and y (as in the variables of this paper) is existentially quantified. Intuitively, each value of x corresponds to a different "slice" of the proof, an idea first used in proof-nets[4, 3], whereas y represents non-determinism, in that we do not know in advance which alternative will lead to a proof.

The key technical issue to combine these rules with the appropriate rule for multiplicatives. One such possibility is the version of $\otimes$R rule below (here illustrated, for simplicity, with only two context formulae).

$$\frac{\vdash F_1[z], F_2[\overline{z}], F_3[x_1.z + \overline{x_1}.\overline{z}], F_4[x_2.z + \overline{x_2}.\overline{z}]}{\vdash F_1 \otimes F_2, F_3, F_4}\ \otimes R$$

where z is universally quantified and x_1, x_2 are existentially quantified.

Note that the two slices in this case are

$$\vdash F_1[1], F_3[x_1], F_4[x_2] \qquad \text{and} \qquad \vdash F_2[1], F_3[\overline{x_1}], F_4[\overline{x_2}]$$

which is precisely what is generated by the techniques of [7] in this case.

Development along these lines will help to form a bridge between sequent systems, which are good for analysis but generally make a poor basis for efficient implementations, and matrix or connection methods [12, 8], on which many efficient implementations are based.

An implementation of the combined system using constraint logic programming techniques with Boolean constraint solver is underway.

Acknowledgements

We thank David Pym and Michael Winikoff for discussions related to this work. The author is grateful for the hospitality of the Department of Computer Science of Queen Mary and Westfield College, University of London during a period of sabbatical leave.

References

1. A. Avron, A Tableaux System for Gödel-Dummett Logic Based on a Hypersequential Calculus, *Proceedings of the 9th International Conference on Analytic Tableaux and Related Methods (TABLEAUX'2000)* 98-111, St. Andrews, July, 2000. Published by Springer as LNAI 1847 (Roy Dyckhoff, ed.).
2. D. Gabbay, *Labelled Deductive Systems: Principles and Applications. Vol 1: Basic Principles*, Oxford University Press, 1996.
3. D. Galmiche and J.M. Notin, Proof-search and Proof nets in Mixed Linear Logic, *Electronic Notes in Theoretical Computer Science* 37, 2000.
4. J.-Y. Girard, Geometry of Interation III: accommodating the additives, in *Advances in Linear Logic*, J-Y. Girard, Y. Lafont and L. Regnier (eds.), Cambridge University Press, 1995.
5. J. Harland, *On Normal Forms and Equivalence for Logic Programs*, Proceedings of the Joint International Conference and Symposium on Logic Programming 146-160, Washington DC, November, 1992.
6. J. Harland, T. Lutovac, M. Winikoff, *Goal-Directed Proof Search in Multiple-Conclusioned Intuitionistic Logic*, Proceedings of the First International Conference on Computational Logic 254-268, London, July, 2000.
7. J. Harland and D. Pym, Resource-distribution via Boolean constraints (extended abstract), Proceedings of the International Conference on Automated Deduction (CADE-14) 222-236, W. McCune (ed.), Townsville, July, 1997. Published by Springer-Verlag as Lecture Notes in Computer Science 1249.
8. C. Kreitz, H. Mantel, J. Otten and S. Schmitt, Connection-based Proof Construction in Linear Logic, Proceedings of the International Conference on Automated Deduction (CADE-14) 207-221, W. McCune (ed.), Townsville, July, 1997. Published by Springer-Verlag as Lecture Notes in Computer Science 1249.
9. Miller D., Nadathur G., Pfenning F., Ščedrov A. *Uniform proofs as a Foundation for Logic Programming*, Annals of Pure and Applied Logic 51:125-157, 1991.
10. E. Ritter, D. Pym and L. Wallen, On the Intuitionistic Force of Classical Search, *Theoretical Computer Science* 232:299-330, 2000.
11. K. Schütte, *Vollständige Systeme modaler und intuitionistischer Logik*, Springer-Verlag, 1968.
12. L. Wallen, *Automated Deduction in Nonclassical Logics*, MIT Press, 1990.