Search Calculi for Classical and Intuitionistic Logic

James Harland

School of Computer Science and Information Technology
RMIT University
GPO Box 2476V
Melbourne 3001, Australia.
Email: jah@cs.rmit.edu.au
WWW: www.cs.rmit.edu.au/∼jah

Abstract. It is well-known that inference rules in the sequent calculus can be interpreted as both *proof construction rules* (i.e. constructing proofs from the leaves towards the root of the tree) and *proof search rules* (i.e. finding proofs by starting at the (supposed) root and working towards the leaves). However, the information used in each case is different: the former constructs larger proofs from smaller ones, whereas the latter constructs search trees, from which, if the search is successful, a proof can be recovered. Thus during search the intermediate stages are at best *partial proofs*. In this paper we explore a variation of the sequent calculus in which search information, in the form of Boolean constraints, is added to each sequent. In particular, we show how this can be done for the sequent calculus LK for classical logic and the multiple-conclusioned sequent LM for intuitionistic logic. In addition, we show how a judicious use of *hypersequents* can improve the search properties of LM.

1 Introduction

Inferences rules in proof systems such as the sequent calculus can be used in two different ways:

- as rules of *deduction*, i.e. constructing larger proofs from smaller, complete, proofs
- as rules of *reduction*, i.e. showing how a search for a proof of a given sequent can be reduced to search for a proof of a simpler sequent or sequents.

For example, consider the ∧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 $\Gamma \vdash \phi \wedge \psi, \Delta$ (and, for that matter, in only one further step). 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 $\Gamma \vdash \psi, \Delta$.

Note that in the first case we take two existing proofs and generate a larger one, whereas in the second case we reduce a sequent to a simpler set of sequents from which is (hopefully) simpler to find a proof. Thus 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.

This duality is perhaps most strikingly illustrated in the Cut rule.

$$\frac{\Gamma \vdash F, \Delta \quad \Gamma, F \vdash \Delta}{\Gamma \vdash \Delta} \; Cut$$

As a deduction rule, this states, in effect, that using intermediate results as lemmas during deduction does not alter the consequence relation. As a reduction rule, this states that in order to prove a sequent $\Gamma \vdash \Delta$ it is sufficient to find proofs of $\Gamma \vdash F, \Delta$ and $\Gamma, F \vdash \Delta$. However as F can be arbitrary (i.e. there is no clear way to determine F from Γ and Δ) , this does not yield a feasible search operation (unlike, for example, the corresponding operation for the rule ∧R above). Hence Cut is a very useful and powerful rule of deduction, but is often ignored as a rule of reduction [1] (a policy whose correctness is guaranteed by Cut-elimination results [8]).

In this paper we perform some initial exploration of this difference between deduction and reduction. In particular, our intention is to explore the possibilites for the development search-oriented inference rules, rather than the more

[1] But not always – when further information is known about F, it is possible to generate feasible search operation on *analytic* cuts.

traditional proof-construction rules of the sequent calculus. To this end, we are interested in ways in which the "low-level" choices made during proof search, such as the particular formula instances used in an axiom or the $\vee$-R rule, can be explicitly represented in the object generated during the search. Once an appropriate representation of this information is found, we will then presumably be able to use this to delete unnecessary formulae and/or rule instances from the proof generated, to allow the combination and analysis of partially completed searches, and to provide a basis for a quantitative comparison of different search strategies. For example, a rule application in which the active formulae are all ultimately unnecessary in the resulting proof indicates a certain lack of success for the search strategy used, and so we can use the number of such "irrelevant" active formulae as a measure of the success of the strategy.

Hence the overall goal is the formulation of inference rules which are amenable to such analysis. In this paper we report on some preliminary investigations into the appropriate representation, one based on Boolean constraints [12, 13], and another based on hypersequents[]. The former involves using 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. Note that an incomplete seach tree, whilst not being able to be directly converted into a proof, still contains useful information.

We show how this approach may be applied to the classical sequent calculus LK[8], and also to the multiple-conclusioned intuitionistic calculus LM[19, 20] and the single-conclusioned intuitionistic calculus LJ[8]. It is well known that LM has better search properties than LJ (largely due to greater possibilities for delaying choices, particular for disjunctions). The search version of LM presented below extends this property a little further, in that the choices for the "special rules", such as $\rightarrow$R below, can be explicitly tracked by the Boolean variables.

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

In the search version of this rule, we allow the formulae in Δ to remain, but explictly marked with a value of 0 (and the subformula of $F_1 \rightarrow F_2$ are marked with 1).

$$\frac{\Gamma, F_1[1] \vdash F_2[1], \Delta[0]}{\Gamma \vdash F_1 \rightarrow F_2, \Delta} \rightarrow \mathrm{R}$$

Another approach is to make use of hypersequents, which may be thought of as introducing a "sequent level" disjunction, and hence the branches in the generated tree can be either the traditional conjunctive type or a disjunction of sequents. Whilst hypersequents are most commonly used in sequent systems for intermediate logics such as Gödel-Dummett logic [3], and hence are usually based on single-conclusioned systems, the idea seems so simple and natural that it is easy to envisage a hypersequential version of any sequent calculus. In particular, we give a hypersequential version of LM, which allows the object generated during search to contain more information about the search choices made than LM.

It should be noted that both methods (i.e. Boolean constraints and hypersequents) may be thought of as blurring the distinction between LJ and LM, i.e. between single- and multiple-conclusioned systems, in order to better reflect choices made during search.

This paper is organised as follows: in Section 2 we give some background on the sequent calculus and in Section 3 we discuss the basic ideas and show how it may be applied to LK. In Section 4 we investigate the case for LM and in Section 5 we do the same for LJ. In Section 6 we explore the relationship between the search version of LM and hypersequents. Finally in Section 7 we present our conclusions.

2 Background

In a sequent $\Gamma \vdash \Delta$, we refer to Γ as the *antecedent* and Δ as the *succedent*. The rules for LK are well known [8] and are given in the Appendix. Note that for convenience, we use the version of the system in which the weakening rules can be omitted due to the form of the axioms (a property which will also hold for LJ and LM). Despite this, it is often convenient to retain the weakening rules explicitly.

The sequent calculus LJ for intuitionistic logic is as for LK above, with the extra constraint that for each succedent in each inference rule, we require that $|\Delta| \leq 1$. This means that the $\rightarrow$L and $\vee$R rules become

$$\frac{\Gamma \vdash F_1 \quad \Gamma, F_2 \vdash \Delta}{\Gamma, F_1 \rightarrow F_2 \vdash \Delta} \rightarrow L \qquad \frac{\Gamma \vdash F_i}{\Gamma \vdash F_1 \vee F_2} \vee R$$

for some $i = 1, 2$. The $\rightarrow$L rule in LJ thus omits the duplication of Δ in the left-hand premise and the $\vee$R rule must choose which of F_1 and F_2 is to appear in the premise. This also has the effect of banning the CR rule, and restricting the WR rule to case when the succedent of the premise is empty.

The sequent calculus LM for intuitionistic logic is as for LK above, with the extra constraint that for the three rules $\forall$R, $\rightarrow$R and $\neg$R, any "unnecessary" formulae in the succedent of the premise are deleted. Thus these rules become

$$\frac{\Gamma \vdash F[y/x]}{\Gamma \vdash \forall x F, \Delta} \forall R \qquad \frac{\Gamma, F_1 \vdash F_2}{\Gamma \vdash F_1 \rightarrow F_2, \Delta} \rightarrow R \qquad \frac{\Gamma, \neg F \vdash}{\Gamma \vdash \neg F, \Delta} \neg R$$

where y is not free in Γ, Δ or F.

To aid in the discussion of such rules, we adopt the following terminology (from [7]). Note that we need, strictly speaking, to refer to formula occurrences rather than formulae; where appropriate we will assume that there is a means of distinguishing formula occurrences from each other.

Definition 1. *The* active formulae *of an inference are the formula occurrences which are present in the premise(s), but not in the conclusion. The* principal *formula occurrence of an inference is the formula occurrence which is present in the conclusion, but not in the premise(s).*

We refer to formula occurrences which are neither active nor principal as the context*.*

An active formula which appears on the left (resp. right) of $\vdash$ is referred to as left-active *(resp.* right-active*).*

A rule which contains a left-active (resp. right-active) formula is referred to as a left-active *(resp.* right-active*) rule.*

For example, in the $\rightarrow$L rule of LK above, the principal formula is $F_1 \rightarrow F_2$, the occurrence of F_1 in the left premise is right-active, and the occurrence of F_2 in the right premise is left-active.

Intuitively, an inference converts the active formulae into the principal formula and leaves the context unchanged (but as discussed in [15], this is sometimes too simplistic).

3 Proof Search with Boolean Constraints

Consider the following LK proof.

$$\cfrac{\cfrac{p, q \vdash q, r \quad p \vdash q, p, r}{p, p \rightarrow q \vdash q, r} \rightarrow L}{p, p \rightarrow q \vdash q \vee r} \vee R$$

In order to give the search interpretation of this same sequence of rules on the same endsequent, we will attach a Boolean expression, as in [12, 13], 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 proof is

$$\cfrac{\cfrac{p[x_1], q[y_1] \vdash q[z_1], r[z_{21}] \quad p[x_2] \vdash p[y_2], r[z_{22}]}{p[x], p \rightarrow q[y] \vdash q[z_1], r[z_2]} \rightarrow L}{p[x], p \rightarrow q[y] \vdash q \vee r[z]} \vee R$$

From the axioms we determine that $x_1 = 0, y_1 = z_1 = 1, z_{21} = z_{22} = 0, x_2 = y_2 = 1$ and thus we have

$$\cfrac{\cfrac{p[0], q[1] \vdash q[1], r[0] \quad p[1] \vdash p[1], r[0]}{p[x], p \rightarrow q[y] \vdash q[z_1], r[z_2]} \rightarrow L}{p[x], p \rightarrow q[y] \vdash q \vee r[z]} \vee R$$

from which we determine that $x = 1, y = 1, z_1 = 1, z_2 = 0, z = 1$, yielding

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

Intuitively, formulae in the endsequent marked with 0 are those which are not necessary in the proof (or at least not up to that point), and hence can be safely deleted. Doing so in the above proof results in the proof below.

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

Note that this proof is slightly more precise than the original one above; in the latter proof it is been detected that p is not required in the left hand branch of →L.

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 [12], which, owing to its basis in linear logic, requires that active formulae be used exactly once; in LK (and LM and LJ for that matter) this restriction does not apply, and so active formulae are not necessarily used in the proof. In this way the use of Boolean constraints here is more akin to a "use check", as in LWB.

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.

Solving the equations then gives us
$w_1 = 1, z_1 = 0, x_{11} = 1, x_{12} = 0, y_1 = 0, w_2 = 0, z_2 = 1, x_{21} = 0, x_{22} = 0, y_2 = 1$
$w = 1, z = 1, x_1 = 1, x_2 = 0, x = 1, y = 1$ [2] and hence we have the "completed" search tree below.

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

Note that as both p and s (in the antecedent) are used on exactly one of the branches of the ∧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 on each branch.

From this completed search tree it is not hard to see that as one of the active formulae of the ∧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 below.

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

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, it is possible to delete from the proof any rule instances in which the active formulae are assigned the

[2] Note that as there is more than one way to make the right-hand leaf into an axiom, there is more than set of equations that can be generated.

value 0. The only exception to this is the rule ∨R, and then only when exactly one of the active formulae is assigned the value 0. In the case of the latter proof above, it is clear that this condition holds for the active formulae of ∨R (i.e. $p[1], q[0]$) which has a clear correspondence with the LJ version of this rule instance, given below.

$$\frac{p, s \vdash p}{p, s \vdash p \vee q} \vee R$$

However, if both active formulae are assigned 0, then any such instance of ∨R can also to be deleted (which will require some simple rearrangement of the proof).

Hence this version of LK allows use to make some refinements to the proofs found. Below we give the formal definitions of this process.

Definition 2. *An* annotated *formula is a formula ϕ together with a Boolean expression e,*[3] *denoted as $\phi[e]$. We denote by $exp(\psi)$ the Boolean expression associated with the annotated formula ψ. Let X be a set of distinct Boolean variables. 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 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 $\phi[1]$; it will always be possible to disambiguate such annotations from the context.

Definition 4. *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 LK.

Definition 5. *An* LK-search derivation *is built via the following rules:*

[3] Here the expressions are just variables. In other contexts (such as [12]) expression are more complex.

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

$$\frac{\Gamma \vdash \Delta}{\Gamma, F[z] \vdash \Delta} \text{ WL} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash F[z], \Delta} \text{ WR}$$

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

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

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

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

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

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

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

where

- $z = 1$ *for each rule other than Axiom* [4]
- *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,* $\to$*L and* $\wedge$*R obey the binary rule search condition and* X', X'', Y' *and* Y'' *are all distinct sets of Boolean variables*

Definition 6. *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 the Axiom rule. A* successful search 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 results. The proofs are straightforward and hence omitted.

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

Proposition 2. *For any formula* F *and any search sequent* $\Gamma \vdash \Delta$*,* $\Gamma \vdash \Delta$ *has an LK successful search tree iff* $\Gamma, F[0] \vdash \Delta$ *has an LK successful search tree iff* $\Gamma \vdash \Delta, F[0]$ *has an LK successful search tree.*

These results are hardly surprising, but they do show how "tighter" LK proofs can be found. We can also use the search calculus to keep track of the progress search strategies. For example, a rule application in which all active formulae are assigned the value 0 is unnecessary, and can be deleted from the proof, and a strategy which leads to a proof which does not contain any such occurrences can be considered more successful than one which does. Moreover, we can quantify this difference by assigning a measure to each proof corresponding to the number of "relevant" active formulae. More on this point is outside the scope of this paper; we will take up this theme in subsequent work.

[4] It is possible make this constraint explicit in each rule. However the version used here makes it simpler to compare systems.

4 Search Calculus for LM

LM uses the same rules as LK except for the rules ∀R, →R and ¬R which are referred to in [20] as the *special rules*. As 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), the search version of LM requires for the special rules 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.

For example, the following instance of the →R rule

$$\frac{p \to r, r \to q, p \vdash q}{p \to r, r \to q \vdash p \to q, \neg r, \forall x s(x)} \to R$$

would be represented in the search calculus as

$$\frac{p \to r[x], r \to q[y], p[z_1] \vdash q[z_2], \neg r[0], \forall x s(x)[0]}{p \to r[x], r \to q[y] \vdash p \to q[z], \neg r[w], \forall x s(x)[v]} \to R$$

This example also serves to point out that the Boolean values are intended to reflect the choice of rule application and principal formula, and not to specify such choices. Hence whatever method is used to make such choices, there will be a corresponding Boolean assignment.

Hence to obtain a search calculus for LM we only need to add the following constraint to those in Definition 5:

– in the succedent of the premise of the rules ∀R, →R and ¬R we have that $\forall x \in \exp(\Delta)\ x = 0$

It is then straightforward to show the following results.

Proposition 3. *$\Gamma \vdash \Delta$ has a proof in LM iff there is a successful search tree of a corresponding search sequent.*

Proposition 4. *For any formula F and any search sequent $\Gamma \vdash \Delta$, $\Gamma \vdash \Delta$ has an LM successful search tree iff $\Gamma, F[0] \vdash \Delta$ has an LM successful search tree iff $\Gamma \vdash \Delta, F[0]$ has an LM successful search tree.*

Given these search rules, it is not hard to show that we can always extract an LJ proof from an LM successful search tree. Whilst this is not a new result[4, 18] and not exactly surprising, the Boolean variables allow a simple statement of this conversion process, as below:

1. Permute all occurrences of ∨L as close to the root as possible
2. Permute all right-active rules as close to the leaves as possible
3. Delete formulae marked 0
4. Delete any rule instances in which at least one active formula is marked 0 (except for ∨R, in which case only delete those in which both active formulae are marked 0)
5. Insert instances of ∨R immediately above any instances of ∨L.

Given any successful proof search in LM, this process will 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 7. *We denote by the* multiplicity *of a LM search tree the maximum number of formulae in a succedent in the search tree whose value is 1.*

Thus we get the result below.

Proposition 5. *If $\Gamma \vdash \Delta$ has an LM successful search 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 successful search 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 ∨L. At such rule instances, any subsequent right-active rules (including →L and ∧R) are either inapplicable or can be permuted upwards, and so any formula in the succedent of the conclusion of ∨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 ∨L into one disjunction (step 5), thus generating a proof of multiplicity 1. □

5 Search Calculus for LJ

Having developed a search calculus for LM, it may seem a little unusual to then develop one for LJ, in that LM allows greater postponement of choices and the above process shows how an LJ proof can be recovered from a successful LM search. However, doing so allows a clear illustration of the difference between the two in terms of proof search, as well as illustrating the generality and simplicity of this approach.

A key feature of LJ is the form of the ∨R rule, i.e.

$$\frac{\Gamma \vdash F_i}{\Gamma \vdash F_1 \vee F_2}\ \vee\mathrm{R}$$

for some $i = 1, 2$.

The natural way to write this rule as a search rule is

$$\frac{\Gamma \vdash F_1[x], F_2[\overline{x}]}{\Gamma \vdash F_1 \vee F_2[1]}\ \vee\mathrm{R}$$

Now as this requires that the succedents contain more than one formula, this may seem merely to lead us back to LM. However, in this case the assignment to the variables is done differently; in LM the corresponding rule is

$$\frac{\Gamma \vdash F_1[x_1], F_2[x_2], \Delta}{\Gamma \vdash F_1 \vee F_2[1], \Delta}\ \vee\mathrm{R}$$

in which there is no relationship between x_1 and x_2. Alternatively the LJ rule is a special case of the LM rule in which $x_1 = \overline{x_2}$ and all variables for formulae in Δ are 0.

For example, consider the LJ proof below.

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

The LJ search version of this is

$$\cfrac{\cfrac{\cfrac{\cfrac{r[x_7],p[x_9] \vdash p[y_3],s[0],q[0] \quad r[x_8],p[x_{10}],s[y_4] \vdash s[w],q[\overline{w}]}{r[x_3],p[x_5],p \to s[y_1] \vdash s[w],q[\overline{w}]}\ \to L}{r[x_3],p[x_5],p \to s[y_1] \vdash s \vee q[z]}\ \vee R \quad \cfrac{r[x_4],q[x_6],p \to s[y_2] \vdash s[v],q[\overline{v}]}{r[x_4],q[x_6],p \to s[y_2] \vdash s \vee q[z]}\ \vee R}{r[x_1], p \vee q[x_2], p \to s[y] \vdash s \vee q[z]}\ \vee L}{r \wedge (p \vee q)[x], p \to s[y] \vdash s \vee q[z]}\ \wedge L$$

where we have

$$x = 1, x_2 = 1, z = 1, y_1 = 1, x_9 = 1, y_3 = 1, x_7 = 0, x_8 = 0, x_{10} = 0$$
$$y_4 = 1, w = 1, x_4 = 0, x_6 = 1, v = 0, y_2 = 0$$

and so the binary search rule condition ensures that

$$x_3 = 0, x_5 = 1, x_1 = 0, x_2 = 1, y_1 = 1, y = 1$$

This results in the search tree below.

$$\cfrac{\cfrac{\cfrac{\cfrac{r[0],p[1] \vdash p[1],s[0],q[0] \quad r[0],p[0],s[1] \vdash s[1],q[0]}{r[0],p[1],p \to s[1] \vdash s[1],q[0]}\ \to L}{r[0],p[1],p \to s[1] \vdash s \vee q[1]}\ \vee R \quad \cfrac{r[0],q[1],p \to s[0] \vdash s[0],q[1]}{r[0],q[1],p \to s[0] \vdash s \vee q[1]}\ \vee R}{r[0], p \vee q[1], p \to s[1] \vdash s \vee q[1]}\ \vee L}{r \wedge (p \vee q)[1], p \to s[1] \vdash s \vee q[1]}\ \wedge L$$

Despite the potential presence of more than one formulae in the succedent, this system is different from LM. In particular, we use a more restricted form of the $\vee$L rule. Consider the provable sequent $p \vee q \vdash p \vee q$. Applying the $\vee$R rule closer to the root than $\vee$L will not result in an LJ proof. Hence we must reflect this by there not being an LJ-search proof of $p \vee q[x] \vdash p[y], q[\overline{y}]$. If we apply the search form of $\vee$R and then $\vee$L from LM to the sequent $p \vee q \vdash p \vee q$ we get

$$\dfrac{\dfrac{p[x_1] \vdash p[y_{11}], q[y_{12}] \quad q[x_2] \vdash p[y_{21}], q[y_{22}]}{p \vee q[x] \vdash p[y_1], q[y_2]}\ \vee\mathrm{L}}{p \vee q[x] \vdash p \vee q[z]}\ \vee\mathrm{R}$$

which is provable giving the completed search proof

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

To prevent this behaviour in LJ, we do not refresh the variables in the succedents of the premises of the $\vee$L rule, but leave them the same as the variables in the succedent of the conclusion. In the above example we would get the following search derivation

$$\dfrac{\dfrac{p[x_1] \vdash p[y], q[\overline{y}] \quad q[x_2] \vdash p[y], q[\overline{y}]}{p \vee q[x] \vdash p[y], q[\overline{y}]}\ \vee\mathrm{L}}{p \vee q[x] \vdash p \vee q[z]}\ \vee\mathrm{R}$$

which does not lead to a successful search, as we cannot simultaniously close both leaves (which requires both $y = 1$ and $\overline{y} = 1$. Thus whilst we allow multiple formulae in succedents, these are not "free" in the same way as in LM.

The rules for LJ are given below.

Definition 8. *An* LJ-search derivation *is built via the following rules:*

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

$$\frac{\Gamma \vdash \Delta}{\Gamma, F[z] \vdash \Delta}\ \text{WL} \qquad \frac{\Gamma \vdash \Delta[0]}{\Gamma \vdash F[z], \Delta[0]}\ \text{WR}$$

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

$$\frac{\Gamma, F_1[x], F_2[y] \vdash \Delta}{\Gamma, F_1 \wedge F_2[z] \vdash \Delta}\ \wedge\text{L} \qquad \frac{\Gamma[X'] \vdash F_1[x], \Delta[0] \quad \Gamma[X''] \vdash F_2[y], \Delta[0]}{\Gamma[X] \vdash F_1 \wedge F_2[z], \Delta[0]}\ \wedge\text{R}$$

$$\frac{\Gamma[X'], F_1[x] \vdash \Delta[Y] \quad \Gamma[X''], F_2[y] \vdash \Delta[Y]}{\Gamma[X], F_1 \vee F_2[z] \vdash \Delta[Y]}\ \vee\text{L} \qquad \frac{\Gamma \vdash F_1[x], F_2[\overline{x}], \Delta[0]}{\Gamma \vdash F_1 \vee F_2[z], \Delta[0]}\ \vee\text{R}$$

$$\frac{\Gamma, F[y/x][w] \vdash \Delta}{\Gamma, \exists x F[z] \vdash \Delta}\ \exists\text{L} \qquad \frac{\Gamma \vdash F[t/x][1], \Delta[0]}{\Gamma \vdash \exists x F[z], \Delta[0]}\ \exists\text{R}$$

$$\frac{\Gamma, F[t/x][w] \vdash \Delta}{\Gamma, \forall x F[z] \vdash \Delta}\ \forall\text{L} \qquad \frac{\Gamma \vdash F[y/x][w], \Delta[0]}{\Gamma \vdash \forall x F[z], \Delta[0]}\ \forall\text{R}$$

$$\frac{\Gamma[X'] \vdash F_1[x_1], \Delta[0] \quad \Gamma[X''], F_2[x_2] \vdash \Delta}{\Gamma[X], F_1 \rightarrow F_2[z] \vdash \Delta} \rightarrow \text{L} \qquad \frac{\Gamma, F_1[x_1] \vdash F_2[x_2], \Delta[0]}{\Gamma \vdash F_1 \rightarrow F_2[z], \Delta[0]} \rightarrow \text{R}$$

$$\frac{\Gamma \vdash F[x], \Delta[0]}{\Gamma, \neg F[z] \vdash \Delta[0]}\ \neg\text{L} \qquad \frac{\Gamma, F[x] \vdash \Delta[0]}{\Gamma \vdash \neg F[z], \Delta[0]}\ \neg\text{R}$$

where

- $z = 1$ *for each rule other than Axiom*
- *the rules ∃L and ∀R have the side condition that* y *is not free in* Γ, Δ *or* F
- *the rules ∨L, →L and ∧R obey the binary rule search condition and* X', X'', Y' *and* Y'' *are all distinct sets of Boolean variables*

That this system is still based on LJ rather than being a variant of LM is shown by the following result.

Proposition 6. *Let* Φ *be an LJ successful search tree with Boolean assignment* I. *Then for every sequent* $\Gamma \vdash \Delta$ *in* Φ, $|\Delta[I]^1| \leq 1$.

In other words, in every LJ search proof, there is at most one formula in every succedent whose Boolean assignment is 1. This makes the following results straightforward.

Proposition 7. $\Gamma \vdash \Delta$ *has a proof in LJ iff there is a successful search tree of a corresponding search sequent.*

Proposition 8. *For any formula* F *and any search sequent* $\Gamma \vdash \Delta$, $\Gamma \vdash \Delta$ *has an LJ successful search tree iff* $\Gamma, F[0] \vdash \Delta$ *has an LJ successful search tree iff* $\Gamma \vdash \Delta, F[0]$ *has an LJ successful search tree.*

The LJ search calculus thus allows more than one formulae to appear in a succedent, as does LM; the difference lies in the manner of how choices become committed. In the LJ case, once a formula in a succedent which has an ancestor formula appearing as an active formula of ∨R closer to the root has been chosen as the principal formula, then all remaining formulae in that succedent cannot be so chosen. In the case of LM, this decision is not dependent on any ancestral relationship, but much more local (and simpler) properties of the special rules.

6 Hypersequents and search

Having explored search issues in LK, LM and LJ, it seems natural to compare the above techniques with that of *hypersequents* [1–3], which may be thought of as introducing a “sequent level” disjunction. This is done by introducing an inter-sequent operator | which intuitively acts as a disjunction:

$$\Gamma_1 \vdash \Delta_1 \mid \Gamma_2 \vdash \Delta_2 \text{ is provable iff } \Gamma_i \vdash \Delta_i \text{ is provable for some } i = 1, 2.$$

From the perspective of proof search, this introduces disjunctive branches into the search tree (which, as per the original sequent calculus, contained only conjunctive branches). Whilst the axioms remain the same, the introduction of (in the terminology of[2]) *external* structural rules means that sequents can be copied (via contraction), have their order changed (via exchange) or deleted (via weakening).

For example, the hypersequent $p, p \to q \vdash q \mid p, p \to q \vdash r$ has the following proof.

$$\dfrac{\dfrac{p \vdash p}{p \vdash p \mid p, p \to q \vdash r}\ EW \quad \dfrac{p, q \vdash q}{p, q \vdash q \mid p, p \to q \vdash r}\ EW}{p, p \to q \vdash q \mid p, p \to q \vdash r} \to \mathrm{L}$$

Such systems are often used to study *intermediate* logics (i.e. those between intuitionistic and classical in strength), such as Gödel-Dummett logic [3], and can be considered a generalisation of LJ which is orthogonal to LK (i.e. multi-conclusioned sequents vs. single-conclusioned hypersequents) [1]. However, the idea seems so simple and natural that it is easy to envisage a hypsequential version of any sequent calculus.

In our context, hypersequents seem to be an elegant way to deal with the special rules in LM. In particular, as it is not known at the time of the application of the rule whether this choice will lead to a proof or not, it is useful to be able to "preserve" the alternatives.

This can be done quite naturally as follows. We change, for instance, the $\to$ rule, below left, into the form below on the right.

$$\dfrac{\Gamma, F_1 \vdash F_2}{\Gamma \vdash F_1 \to F_2, \Delta} \to R \qquad\qquad \dfrac{\Gamma, F_1 \vdash F_2 \mid \Gamma \vdash \Delta}{\Gamma \vdash F_1 \to F_2, \Delta} \to R$$

Below we give the full set of rules for such a hypersequential version of LM. For simplicity, we consider a hypersequent to be a multi-set of sequents, in order to dispense with the external exchange rule (which is an analogue of common practice for standard sequent systems). We use G and H to refer to multi-sets of sequents, and S to refer to an individual sequent. Note also that for the binary rules, we may assume additive behaviour of the "extra" sequents, due to the presence of external weakening and contraction.

Definition 9. *We define the rules of the hypersequential variant of LM, denoted em LMH as follows:*

$$\frac{G}{G \mid H}\ EW \qquad \frac{S \mid H}{S \mid S \mid H}\ EC$$

$$\frac{}{\Gamma, F \vdash F, \Delta}\ \mathrm{Axiom} \qquad \frac{\Gamma \vdash F, \Delta \mid H \quad \Gamma, F \vdash \Delta \mid H}{\Gamma \vdash \Delta \mid H}\ \mathrm{Cut}$$

$$\frac{\Gamma \vdash \Delta \mid H}{\Gamma, F \vdash \Delta \mid H}\ \mathrm{WL} \qquad \frac{\Gamma \vdash \Delta \mid H}{\Gamma \vdash F, \Delta \mid H}\ \mathrm{WR}$$

$$\frac{\Gamma, F, F \vdash \Delta \mid H}{\Gamma, F \vdash \Delta \mid H}\ \mathrm{CL} \qquad \frac{\Gamma \vdash F, F, \Delta \mid H}{\Gamma \vdash F, \Delta \mid H}\ \mathrm{CR}$$

$$\frac{\Gamma, F_1, F_2 \vdash \Delta \mid H}{\Gamma, F_1 \wedge F_2 \vdash \Delta \mid H}\ \wedge\mathrm{L} \qquad \frac{\Gamma \vdash F_1, \Delta \mid H \quad \Gamma \vdash F_2, \Delta \mid H}{\Gamma \vdash F_1 \wedge F_2, \Delta \mid H}\ \wedge\mathrm{R}$$

$$\frac{\Gamma, F_1 \vdash \Delta \mid H \quad \Gamma, F_2 \vdash \Delta \mid H}{\Gamma, F_1 \vee F_2 \vdash \Delta \mid H}\ \vee\mathrm{L} \qquad \frac{\Gamma \vdash F_1, F_2, \Delta \mid H}{\Gamma \vdash F_1 \vee F_2, \Delta \mid H}\ \vee\mathrm{R}$$

$$\frac{\Gamma \vdash F, \Delta \mid H}{\Gamma, \neg F \vdash \Delta \mid H}\ \neg\mathrm{L} \qquad \frac{\Gamma, F \vdash\ \mid \Gamma \vdash \Delta \mid H}{\Gamma \vdash \neg F, \Delta \mid H}\ \neg\mathrm{R}$$

$$\frac{\Gamma \vdash F_1, \Delta \mid H \quad \Gamma, F_2 \vdash \Delta \mid H}{\Gamma, F_1 \to F_2 \vdash \Delta \mid H}\ \to\mathrm{L} \qquad \frac{\Gamma, F_1 \vdash F_2 \mid \Gamma \vdash \Delta \mid H}{\Gamma \vdash F_1 \to F_2, \Delta \mid H}\ \to\mathrm{R}$$

$$\frac{\Gamma, F[t/x] \vdash \Delta \mid H}{\Gamma, \forall x F \vdash \Delta \mid H}\ \forall\mathrm{L} \qquad \frac{\Gamma \vdash F[y/x] \mid \Gamma \vdash \Delta \mid H}{\Gamma \vdash \forall x F, \Delta \mid H}\ \forall\mathrm{R}$$

$$\frac{\Gamma, F[y/x] \vdash \Delta \mid H}{\Gamma, \exists x F \vdash \Delta \mid H}\ \exists\mathrm{L} \qquad \frac{\Gamma \vdash F[t/x], \Delta \mid H}{\Gamma \vdash \exists x F, \Delta \mid H}\ \exists\mathrm{R}$$

where the rules $\exists$*L and* $\forall$*R have the side condition that* y *is not free in* Γ, Δ *or* F.

Note that when viewed as a search calculus (i.e. we construct trees from the root upwards), the only rules which increase the number of sequents are the special rules. Hence any proof of a (single) sequent $\Gamma \vdash \Delta$ in the above system which does not contain the special rules is just an LM proof (an , indeed, an LK proof).

In fact is is not hard to show the following result.

Proposition 9. *Let* H *be a hypersequent provable in LMH. Then there is a sequent* $S \in H$ *such that* S *is provable in LMH.*

The proof is a simple induction on the size of the proof and is not given here.

Hence any LMH proof can be transformed into an LM proof of one of the sequents in the hypersequent at the root. In addition, it is simple enough to see that any LM proof can be transformed into an LMH proof.

As for LJ, a natural use of hypersequents is for the $\vee$R rule, i.e.

$$\frac{\Gamma \vdash F_1 \mid \Gamma \vdash F_2 \mid H}{\Gamma \vdash F_1 \vee F_2 \mid H}$$

which, in fact, is already used in the GLC* system for the Gödel-Dummett logic LC [3]. This approach has the advantage of maintaining the single-conclusioned system.

7 Conclusions and Further Work

We have shown how Boolean constraints can be used, in the manner of [12, 13] to develop search calculi for LK, LM and LJ. The main contribution here is not so much the technical results, but the general approach to the development of such calculi by a simple technique, and for which the implementation issues are straightforward (in that there are a number of readily available software packages for solving Boolean constraints).

Further investigation into the relationship between such search calculi and hypersequents appears to be a promising direction for future work. The connection between hypersequents and Gödel-Dummett logic is somewhat tantalizing; in [10] 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 [16], amongst others) in the derivation of logic programming languages and some more recent work on similar analyses in LM [11] suggests that a natural conclusion is to perform analyses of logic programming languages in a search calculus for Gödel-Dummett logic. The hypersequential version of LM above seems to be a good starting point for such an investigation (or indeed the GLC* system of [3]).

A further point of interest is to extend the original system of [12, 13] 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. Hence instead of generating implicitly existentially quantified constraints such as $x = 1, y = z$, we would generate $\forall x C_1$ and $\exists y C_2$ respectively, where each C_i is the constraint (possibly further quantified) generated by the rest of the proof search.

Intuitively, each value of x corresponds to a different "slice" of the proof, an idea first used in proof-nets[9, 6], 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 [12] 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 [20, 14], 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, Michael Winikoff and Tatjana Lutovac 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, Two Types of Multiple-Conclusion Systems, *Logic Journal of the IGPL* 6:5:695-717, 1998.
2. 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.).
3. A. Avron and B. Konikowska, Decomposition Proof Systems for Gödel Logics, *Studia Logica* 69:197-219, 2001.

4. U. Egly and S. Schmitt, On Intuitionistic Proof Transformations, Their Complexity and Application to Constructive Program Synthesis, *Fundamenta Informaticae* 39:59-83, 1999.
5. D. Gabbay, *Labelled Deductive Systems: Principles and Applications. Vol 1: Basic Principles*, Oxford University Press, 1996.
6. D. Galmiche and J.M. Notin, Proof-search and Proof nets in Mixed Linear Logic, *Electronic Notes in Theoretical Computer Science* 37, 2000.
7. D. Galmiche and G. Perrier, On proof normalization in Linear Logic, *Theoretical Computer Science* 135, 67-110, 1994.
8. G. Gentzen, Untersuchungen über das logische Schliessen, *Math. Zeit.* 39, 176-210, 405-431, 1934.
9. 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.
10. 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.
11. 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.
12. 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.
13. J. Harland and D. Pym, Resource-distribution via Boolean constraints, to appear in the ACM Transactions on Computational Logic.
14. 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.
15. Lutovac T., Harland J. *Towards the Automation of the Design of Logic Programming Languages*, Technical Report TR-97-30, Department of Computer Science, RMIT University, 1997. `http://www.cs.rmit.edu.au/~jah/publications/perms-tr.ps.gz`.
16. 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.
17. E. Ritter, D. Pym and L. Wallen, On the Intuitionistic Force of Classical Search, *Theoretical Computer Science* 232:299-330, 2000.
18. S. Schmitt, L. Lorigo, C. Kreitz and A. Nogin, `JProver:` Integrating Connection-based Theorem Proving into Interactive Proof Assistants, Proceedings of the International Joint Conference on Automated Deduction 421-5, Siena, June, 2001.
19. K. Schütte, *Vollständige Systeme modaler und intuitionistischer Logik*, Springer-Verlag, 1968.
20. L. Wallen, *Automated Deduction in Nonclassical Logics*, MIT Press, 1990.

A Sequent Calculus LK

$$\frac{}{\Gamma, F \vdash F, \Delta}\ \text{Axiom} \qquad \frac{\Gamma \vdash F, \Delta \quad \Gamma, F \vdash \Delta}{\Gamma \vdash \Delta}\ \text{Cut} \qquad \frac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}\ \text{WL} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash F, \Delta}\ \text{WR}$$

$$\frac{\Gamma, F, F \vdash \Delta}{\Gamma, F \vdash \Delta}\ \text{CL} \qquad \frac{\Gamma \vdash F, F, \Delta}{\Gamma \vdash F, \Delta}\ \text{CR} \qquad \frac{\Gamma, F_1, F_2 \vdash \Delta}{\Gamma, F_1 \wedge F_2 \vdash \Delta}\ \wedge\text{L} \qquad \frac{\Gamma \vdash F_1, \Delta \quad \Gamma \vdash F_2, \Delta}{\Gamma \vdash F_1 \wedge F_2, \Delta}\ \wedge\text{R}$$

$$\frac{\Gamma, F_1 \vdash \Delta \quad \Gamma, F_2 \vdash \Delta}{\Gamma, F_1 \vee F_2 \vdash \Delta}\ \vee\text{L} \qquad \frac{\Gamma \vdash F_1, F_2, \Delta}{\Gamma \vdash F_1 \vee F_2, \Delta}\ \vee\text{R} \qquad \frac{\Gamma \vdash F, \Delta}{\Gamma, \neg F \vdash \Delta}\ \neg\text{L} \qquad \frac{\Gamma, F \vdash \Delta}{\Gamma \vdash \neg F, \Delta}\ \neg\text{R}$$

$$\frac{\Gamma \vdash F_1, \Delta \quad \Gamma, F_2 \vdash \Delta}{\Gamma, F_1 \rightarrow F_2 \vdash \Delta} \rightarrow \text{L} \qquad \frac{\Gamma, F_1 \vdash F_2, \Delta}{\Gamma \vdash F_1 \rightarrow F_2, \Delta} \rightarrow \text{R} \qquad \frac{\Gamma, F[t/x] \vdash \Delta}{\Gamma, \forall x F \vdash \Delta}\ \forall\text{L} \qquad \frac{\Gamma \vdash F[y/x], \Delta}{\Gamma \vdash \forall x F, \Delta}\ \forall\text{R}$$

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

where the rules ∃L and ∀R have the side condition that y is not free in Γ, Δ or F.

B Sequent Calculus LJ

The sequent calculus LJ for intuitionistic logic is as for LK above, with the extra constraint that for each succedent in each inference rule, we require that $|\Delta| \leq 1$. This leads to the rules below.

$$\frac{}{\Gamma, F \vdash F}\ \text{Axiom} \qquad \frac{\Gamma \vdash F \quad \Gamma, F \vdash \Delta}{\Gamma \vdash \Delta}\ \text{Cut} \qquad \frac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}\ \text{WL} \qquad \frac{\Gamma \vdash}{\Gamma \vdash F}\ \text{WR}$$

$$\frac{\Gamma, F, F \vdash \Delta}{\Gamma, F \vdash \Delta}\ \text{CL} \qquad \frac{\Gamma, F_1, F_2 \vdash \Delta}{\Gamma, F_1 \wedge F_2 \vdash \Delta}\ \wedge\text{L} \qquad \frac{\Gamma \vdash F_1 \quad \Gamma \vdash F_2}{\Gamma \vdash F_1 \wedge F_2}\ \wedge\text{R}$$

$$\frac{\Gamma, F_1 \vdash \Delta \quad \Gamma, F_2 \vdash \Delta}{\Gamma, F_1 \vee F_2 \vdash \Delta}\ \vee\text{L} \qquad \frac{\Gamma \vdash F_1, F_2}{\Gamma \vdash F_1 \vee F_2}\ \vee\text{R} \qquad \frac{\Gamma \vdash F}{\Gamma, \neg F \vdash}\ \neg\text{L} \qquad \frac{\Gamma, F \vdash}{\Gamma \vdash \neg F}\ \neg\text{R}$$

$$\frac{\Gamma \vdash F_1 \quad \Gamma, F_2 \vdash \Delta}{\Gamma, F_1 \to F_2 \vdash \Delta} \to \text{L} \qquad \frac{\Gamma, F_1 \vdash F_2}{\Gamma \vdash F_1 \to F_2} \to \text{R} \qquad \frac{\Gamma, F[t/x] \vdash \Delta}{\Gamma, \forall x F \vdash \Delta}\ \forall\text{L} \qquad \frac{\Gamma \vdash F[y/x]}{\Gamma \vdash \forall x F}\ \forall\text{R}$$

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

where the rules ∃L and ∀R have the side condition that y is not free in Γ, Δ or F.

C Sequent Calculus LM

The sequent calculus LM for intuitionistic logic is as for LK above, with the extra constraint that for the three rules ∀R, →R and ¬R, the succedent of the premise is either empty (for ¬R) or contains only active formulae of the rule (∀R, →R).

$$\frac{}{\Gamma, F \vdash F, \Delta}\ \text{Axiom} \qquad \frac{\Gamma \vdash F, \Delta \quad \Gamma, F \vdash \Delta}{\Gamma \vdash \Delta}\ \text{Cut} \qquad \frac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}\ \text{WL} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash F, \Delta}\ \text{WR}$$

$$\frac{\Gamma, F, F \vdash \Delta}{\Gamma, F \vdash \Delta}\ \text{CL} \qquad \frac{\Gamma \vdash F, F, \Delta}{\Gamma \vdash F, \Delta}\ \text{CR} \qquad \frac{\Gamma, F_1, F_2 \vdash \Delta}{\Gamma, F_1 \wedge F_2 \vdash \Delta}\ \wedge\text{L} \qquad \frac{\Gamma \vdash F_1, \Delta \quad \Gamma \vdash F_2, \Delta}{\Gamma \vdash F_1 \wedge F_2, \Delta}\ \wedge\text{R}$$

$$\frac{\Gamma, F_1 \vdash \Delta \quad \Gamma, F_2 \vdash \Delta}{\Gamma, F_1 \vee F_2 \vdash \Delta}\ \vee\text{L} \qquad \frac{\Gamma \vdash F_1, F_2, \Delta}{\Gamma \vdash F_1 \vee F_2, \Delta}\ \vee\text{R} \qquad \frac{\Gamma \vdash F, \Delta}{\Gamma, \neg F \vdash \Delta}\ \neg\text{L} \qquad \frac{\Gamma, F \vdash}{\Gamma \vdash \neg F, \Delta}\ \neg\text{R}$$

$$\frac{\Gamma \vdash F_1, \Delta \quad \Gamma, F_2 \vdash \Delta}{\Gamma, F_1 \to F_2 \vdash \Delta} \to \text{L} \qquad \frac{\Gamma, F_1 \vdash F_2}{\Gamma \vdash F_1 \to F_2, \Delta} \to \text{R} \qquad \frac{\Gamma, F[t/x] \vdash \Delta}{\Gamma, \forall x F \vdash \Delta}\ \forall\text{L} \qquad \frac{\Gamma \vdash F[y/x]}{\Gamma \vdash \forall x F, \Delta}\ \forall\text{R}$$

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

where the rules ∃L and ∀R have the side condition that y is not free in Γ, Δ or F.