A Proof-Theoretic Analysis of Goal-Directed Provability

James Harland
Department of Computer Science
University of Melbourne
Parkville, 3052
Australia
e-mail: jah@cs.mu.oz.au

June, 1993

Abstract

Uniform proofs have been presented as a basis for logic programming, and it is known that by restricting the class of formulae it is possible to guarantee that uniform proofs are complete with respect to provability in intuitionistic logic. In this paper we explore the relationship between uniform proofs and classes of formulae more deeply. Firstly we show that uniform proofs arise naturally as a normal form for proofs in first-order intuitionistic sequent calculus. Next we show that the class of formulae known as hereditary Harrop formulae are intimately related to uniform proofs, and that we may extract such formulae from uniform proofs in two different ways. We also give results which may be interpreted as showing that hereditary Harrop formulae are the largest class of formulae for which uniform proofs are guaranteed to be complete. Finally we briefly discuss some possibilities for a slightly more general approach using intermediate and infinitary logics.

Keywords: proof theory, logic programming, intuitionistic logic, uniform proofs, definiteness.

1 Introduction

It has long been known that there are efficient implementation techniques which make Horn clauses, a particular fragment of first-order logic, able to be used as a programming language [20], and that this class of formulae form the semantic basis for the programming language Prolog [4, 32]. It has also been shown that this computational paradigm is as powerful as that of Turing machines [33]. Thus we may think of Horn clauses as incorporating some form of algorithmic knowledge. As Horn clauses are not a particularly large fragment of first-order logic, it is perhaps not surprising that this class of formulae has such a relatively strong property. There have been various schemes proposed for logic programming languages which

are extensions of Horn clauses [2, 8, 9, 22, 23, 28, 29, 31]. Given these various extensions, it seems natural to ask whether there is a maximal class of formulae which may be used as a programming language. Moreover, there does not seem to be universally agreed criterion which may be used to determine what constitutes a logic programming language, without which any such notion of maximality would seem premature. A criterion of this nature has been proposed by Miller et al. [28], in that they identify various first-order and higher-order fragments as logic programming languages by showing that these fragments satisfy a completeness property for a certain class of proofs. However it would seem that a general criterion should be strong enough not only to verify that certain fragments may be used as programming languages, but also to discover such fragments in the first place. Thus it seems natural to use the criterion of [28], namely the completeness of *goal-directed provability*, to investigate this question of maximality for (first-order) logic programming languages.

An important property of logic programming systems seems to be the notion of an answer substitution, in that if a query G is asked of a program P, then we are not merely interested in whether G is true or not, but also which instances of G are true. To quote from [20], p.11: "In fact, the ideal view of a logic programming system is that it is a black box for computing bindings and our only interest is in its input-output behaviour". Thus the variables of the goal provide the notion of an input and output, and hence we wish to know not merely that $p(x)$ is true, but the terms t for which $p(t)$ is true.

This immediately suggests that we are interested in constructive proofs, for which a natural starting place is intuitionistic logic [7, 18, 10, 21]. This accords well with the work of Miller et al. [28], in that in all their logic programming languages, a goal is derivable iff it is provable intuitionistically. A well-known property of intuitionistic logic is that proofs of theoremhood are constructive, i.e. that $\vdash_I \exists x F \Leftrightarrow \vdash_I F[t/x]$ for some term t, and that $\vdash_I F_1 \vee F_2 \Leftrightarrow \vdash_I F_1$ or $\vdash_I F_2$. These are often referred to as the *existential* property and the *disjunctive* property respectively. Such properties seem precisely what is desired by the above notion of input-output behaviour. Indeed, there is growing interest in the application of constructive logics to computer science in general [24, 16, 5].

When applying intuitionistic logic to programming tasks, we are usually more interested in *consequence* than theoremhood, in that it is not so important to determine whether a given formula is true, but *whether a given deduction is valid*. For example, the intuitionistic theorem $p(a) \supset \exists x p(x)$ is of less interest in this context than the deduction $p(a) \vdash_I \exists x p(x)$. For these reasons it may be interesting to investigate the application of the existential and disjunctive properties to consequences, rather than just theoremhood, so that we require that $\Gamma \vdash \exists x F \Leftrightarrow \Gamma \vdash F[t/x]$ and $\Gamma \vdash F_1 \vee F_2 \Leftrightarrow \Gamma \vdash F_1$ or $\Gamma \vdash F_2$ for any (possibly empty) set of formulae Γ. Note that intuitionistic provability alone does not guarantee this property, as $\exists x p(x) \vdash_I \exists x p(x)$, but clearly there is no instance $p(t)$ of $p(x)$ such that $\exists x p(x) \vdash_I p(t)$. Similar remarks apply to $p \vee q \vdash_I p \vee q$. Hence, in order to investigate constructive consequence, rather than constructive theoremhood, we need to examine the notion of proof rather carefully.

One interesting idea in this context is to look at sub-formulae of the formula we are trying to establish. The existential and disjunctive properties for consequence may be thought of as stating that subformulae of the conclusion must be used in the proof. This idea of using sub-formulae to find proofs may be extended to other formulae as well, and this is the basis

for the class of proofs called *uniform proofs* [28]. A uniform proof is one which the principal connective of the formula is introduced in the last step of the proof; in other words, when searching for a proof of a given formula, we need only consider the immediate subformulae of the desired conclusion. Hence we may think of uniform proofs as *goal-directed*, in that when searching for a uniform proof of a given goal, we may use the structure of the goal to determine the structure of the proof. We will denote uniform provability by $\vdash_u$. Such proofs lead to an identification of formulae with operations in a search space, and hence have a natural interpretation as instructions, thus establishing a direct relationship between proof and computation. This restriction also allows a more feasible implementation of the proof search process than may be done in the case of arbitrary (intuitionistic) proofs. Thus uniform proofs are fundamental to the process of finding constructive consequences.

Uniform proofs may be used as a basis for logic programming [28, 25], and there are several interesting investigations along these lines. As mentioned above, it is not the case that all intuitionistic proofs are uniform. Hence, one of the features of this approach is that the formulae involved are restricted to a class for which uniform proofs are complete with respect to intuitionistic logic, i.e. $F_1 \vdash_I F_2$ iff $F_1 \vdash_u F_2$ when F_1 and F_2 belong to a certain class of formulae. One such restriction is that the formulae which may be used as assertions (i.e. those which may appear on the left of $\vdash$) are Horn clauses, and the goals (i.e. those formulae which may appear on the right of $\vdash$) are conjunctions of atoms, so that when F_1 is a set of Horn clauses and F_2 is a conjunction of atoms, then $F_1 \vdash_I F_2$ iff $F_1 \vdash_u F_2$. Larger classes of formulae for which the existence of uniform proofs is guaranteed may also be given [28], and the largest class of first-order formulae for which this property has been established is known as *hereditary Harrop formulae*. Intuitively, these formulae may be thought of as those which contain no negative occurrences of $\exists$ or $\vee$.

In this paper we examine the relationship between uniform proofs and hereditary Harrop formulae, and we give several results which may be interpreted as establishing the maximality of this class of formulae. As uniformity is a property of proofs rather than formulae, it is not strictly possible to establish that a given class of formulae is the largest one for which uniform proofs are complete. For example, given that $F_1 \vdash_u F_2$, where F_1 and F_2 satisfy some restriction, for any formula F whatsoever, $F_1, F \vdash_u F_2$. However, as we shall see, there is a natural relationship between uniform proofs and hereditary Harrop formulae. Essentially this is that whilst $F_1, F \vdash_u F_2$, there is a hereditary Harrop formula D such that $D \vdash_u F_2$, and D is related to F and F_1 in such a way that D is the formula "doing the work" in the uniform proof. The relationship between the formulae is made precise in later sections.

An important insight which arises from this analysis is that the class of hereditary Harrop formulae arises naturally out of the *permutation properties* of the rules of intuitionistic logic [6, 19]. This occurs by determining when it is possible to permute certain combinations of inference rules so that an arbitrary proof may be converted into a uniform proof. Thus we may identify hereditary Harrop formulae as a logic programming language purely from the notion of a uniform proof and the proof theory of intuitionistic logic; no prior knowledge of logic programming languages *per se* is needed. This suggests that the strategy of studying permutation rules in order to investigate the completeness of goal-directed provability may used to identify logic programming languages independently of the logic in use; such a strategy has been used to identify logic programming languages in linear logic [14, 15, 17].

In this way we may think of the permutation properties of the proof theory of the logic in question (in conjunction with the notion of goal-directed search) as determining what fragments of the logic may be used as a logic programming language.

An important feature to note is that this approach uses only proof theory. This seems appropriate, in that the inferences rules used to construct proofs should be the place where the properties pertinent to mechanization should be most apparent. This is not to say that model-theoretic methods cannot be used to derive similar results; however, despite the proofs occasionally becoming tedious, it seems reasonable to remain within proof theory alone when looking for an inherently syntactic (i.e. automatable) property, such as the ability to serve as a programming language. Model theory may be able to give some deeper insights once the language/s have been found, but for identifying and establishing the basic programming properties of any such programming language, one should presumably use proof theory alone.

The rest of this paper is organised as follows. In section 2 we provide the necessary technical preliminaries, and in the following section we prove some basic results concerning uniform proofs. In section 4 we show how definite formulae may be extracted from a certain type of proof, and in section 5 we show how a different method of extraction may be used for uniform proofs. Finally we discuss some limitations of uniform proofs and some possibilities for further work.

2 Preliminaries

First we define hereditary Harrop formulae. Applications of these formulae to logic programming may be found in [28, 25, 26, 29, 30].

We assume the existence of a finite set of constant and function symbols, and a countable set of variables.

DEFINITION 2.1 *We refer to the set of all ground terms as the* Herbrand universe*, denoted by* $\mathcal{U}$*.*

We refer to the set of all terms (not necessarily ground) as the general Herbrand universe*, denoted by* $\mathcal{U}'$*.*

DEFINITION 2.2 *D and G formulae satisfy*

$$D := A \mid \forall x D \mid D_1 \wedge D_2 \mid G \supset A$$
$$G := A \mid \exists x G \mid \forall x G \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid D \supset G$$

where A is an atomic formula.

We refer to D formulae as definite *formulae, and to G formulae as* goal *formulae. The set of all definite formulae will be referred to as $\mathcal{D}$, and the set of all goal formulae as $\mathcal{G}$.*

A program *is a set of closed definite formulae, and a goal is any closed goal formula.*

Note that a class of formulae with the same expressive power may be given as follows:

$$D := A \mid \forall x D \mid D_1 \wedge D_2 \mid G \supset D$$
$$G := A \mid \exists x G \mid \forall x G \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid D \supset G$$

This will sometimes be more convenient for stating results. It should be clear that any formula in the latter class may be transformed to an equivalent one in the former class, and hence we may use the two interchangeably. The transformation is based on the (intuitionistic) equivalences

$$G \supset (D_1 \wedge D_2) \equiv (G \supset D_1) \wedge (G \supset D_2)$$
$$G \supset (\forall x D) \equiv \forall x (G \supset D)$$

provided that x does not appear free in G.

Note that we do not allow negations in hereditary Harrop formulae. We will refer to the formulae which do not contain any negations as *negation-free* formulae.

In [28] it is shown how an operational notion of proof may be given for the above programs and goals. Below we give a slightly different presentation.

DEFINITION 2.3 *Let D be a definite formula. Then we define*

$$\begin{array}{ll} |A| & = \{A\} \\ |\forall x D| & = \bigcup_{t \in \mathcal{U}'} |D[t/x]| \\ |D_1 \wedge D_2| & = |D_1| \cup |D_2| \\ |G \supset A| & = \{G \supset A\} \end{array}$$

where $\mathcal{U}'$ is the general Herbrand universe.

We also define $|\{D_1, \ldots D_n\}| = \bigcup_{i=1}^{n} |D_i|$.

Let P be a set of definite formulae. Then we define $\vdash_o$ as the least relation satisfying:

- *$P \vdash_o A$ iff $A \in |P|$ or $\exists G \supset A \in |P|$ such that $P \vdash_o G$*
- *$P \vdash_o G_1 \vee G_2$ iff $P \vdash_o G_1$ or $P \vdash_o G_2$*
- *$P \vdash_o G_1 \wedge G_2$ iff $P \vdash_o G_1$ and $P \vdash_o G_2$*
- *$P \vdash_o \exists x G$ iff $P \vdash_o G[t/x]$ for some $t \in \mathcal{U}'$*
- *$P \vdash_o \forall x G$ iff $P \vdash_o G[y/x]$ where y does not occur free in G or P*
- *$P \vdash_o D' \supset G$ iff $P \cup \{D'\} \vdash_o G$*

Another point to note is that we may rename bound variables so that there is no overlap between a given free variable and the bound variables of the formula, i.e. that a given variable x either appears free in a formula F, or does not appear at all in F. For example, asking whether x appears free in $\forall x p(x)$ is the same as asking whether x appears in $\forall y p(y)$.

The major difference between this presentation and that of [28] is that the atomic case is here made explicit. This requires a slight but not significant change to the way in which universally quantified goals are handled. In [28], a new constant is substituted for the universally quantified goal. Whilst this coincides with common practice in proof theory, it makes an explicit statement of the success of atomic goals overly complicated, in that we need to either explicitly incorporate the new constants into the definition of $|D|$, or to allow

for unification of an atomic goal with the head of a clause. The above method seems to be an elegant compromise between these two extremes, but the precise choice of method is not of great importance.

It was shown in [28] that $P \vdash_o G$ iff $P \vdash_I G$ where $\vdash_I$ denotes intuitionistic provability, so that $P \vdash_o G$ iff there is a proof in intuitionistic logic of the sequent $P \longrightarrow G$. The rules for the standard sequent calculus for intuitionistic logic are given below (these are taken from [28]). Note that we are only interested in cut-free proofs; hence we omit the cut rule from the proof system below.

$$\frac{\Gamma, B, C \longrightarrow F}{\Gamma, B \wedge C \longrightarrow F} \wedge\text{-L} \qquad \frac{\Gamma \longrightarrow B \quad \Gamma \longrightarrow C}{\Gamma \longrightarrow B \wedge C} \wedge\text{-R}$$

$$\frac{\Gamma, B \longrightarrow F \quad \Gamma, C \longrightarrow F}{\Gamma, B \vee C \longrightarrow F} \vee\text{-L} \qquad \frac{\Gamma \longrightarrow B}{\Gamma \longrightarrow B \vee C} \; \frac{\Gamma \longrightarrow C}{\Gamma \longrightarrow B \vee C} \vee\text{-R}$$

$$\frac{\Gamma \longrightarrow B \quad \Gamma, C \longrightarrow F}{\Gamma, B \supset C \longrightarrow F} \supset\text{-L} \qquad \frac{\Gamma, B \longrightarrow C}{\Gamma \longrightarrow B \supset C} \supset\text{-R}$$

$$\frac{\Gamma, B[t/x] \longrightarrow F}{\Gamma, \forall x B \longrightarrow F} \forall\text{-L} \qquad \frac{\Gamma \longrightarrow B[y/x]}{\Gamma \longrightarrow \forall x B} \forall\text{-R}$$

$$\frac{\Gamma, B[y/x] \longrightarrow F}{\Gamma, \exists x B \longrightarrow F} \exists\text{-L} \qquad \frac{\Gamma \longrightarrow B[t/x]}{\Gamma \longrightarrow \exists x B} \exists\text{-R}$$

$$\frac{\Gamma \longrightarrow \bot}{\Gamma \longrightarrow B} \bot\text{-R}$$

The rules $\forall$-R and $\exists$-L have the side condition that y is not free in Γ, B or F.

We will often refer to the rules $\wedge$-L, $\vee$-L, $\exists$-L, $\forall$-L and $\supset$-L as left rules, and the rules $\wedge$-R, $\vee$-R, $\exists$-R, $\forall$-R and $\supset$-R as right rules.

An *initial sequent* is a sequent $\Gamma \longrightarrow F$ where F is either an atomic formula or $\bot$ and $F \in \Gamma$. A *proof* for the sequent $\Gamma \longrightarrow F$ is a finite tree, constructed using the above rules, whose root is $\Gamma \longrightarrow F$ and whose leaves are initial sequents.

As is done in [25], we omit the interchange and contraction rules by considering the antecedents of sequents to be sets.[1] Note also that thinning is not necessary due to the way an initial sequent is defined.

The $\bot$-R rule will be of little interest, as we will not be dealing with formulae which may contain $\bot$. This is due to the form of the $\bot$-R rule, in that it does not seem possible to interpret it in a goal-directed manner, as no sub-formula of the formula B appears in the premise, and hence it seems difficult to include this rule and maintain the completeness of

[1] The succedent can only be a singleton set, and so we write succedents as single formulae.

goal-directed provability. Without this rule, $\bot$ is merely another atom, as there is no way to infer anything particular about it, and so we choose to dispense with it.

As the above definition of provability is dependent on the formulae obeying a particular restriction, we give a separate definition of uniformity below.

DEFINITION 2.4 *We define the* uniform rule *for a formula F as follows:*

- *The uniform rule for $F_1 \wedge F_2$ is* $\wedge$-R
- *The uniform rule for $F_1 \vee F_2$ is* $\vee$-R
- *The uniform rule for $\exists x F$ is* $\exists$-R
- *The uniform rule for $\forall x F$ is* $\forall$-R
- *The uniform rule for $F_1 \supset F_2$ is* $\supset$-R

We say that a formula F is compound *if F is not an atom.*

DEFINITION 2.5 *A proof Ξ is* uniform *if for each non-initial sequent $\Gamma \longrightarrow F$ in Ξ where F is a compound formula, the rule used to derive $\Gamma \longrightarrow F$ is the uniform rule for F.*

It should be clear that the following proposition holds.

PROPOSITION 2.6 *Let F be a formula containing no negations, and let Γ be a set of such formulae.*

Then

- $\Gamma \vdash_u F_1 \vee F_2$ *iff* $\Gamma \vdash_u F_1$ *or* $\Gamma \vdash_u F_2$
- $\Gamma \vdash_u F_1 \wedge F_2$ *iff* $\Gamma \vdash_u F_1$ *and* $\Gamma \vdash_u F_2$
- $\Gamma \vdash_u \exists x F$ *iff* $\Gamma \vdash_u F[t/x]$ *for some* $t \in \mathcal{U}'$
- $\Gamma \vdash_u \forall x F$ *iff* $\Gamma \vdash_u F[y/x]$ *where y is not free in Γ or F*
- $\Gamma \vdash_u F_1 \supset F_2$ *iff* $\Gamma, F_1 \vdash_u F_2$

If P is a program and G is a goal, then it is clear that $P \vdash_o G$ implies that $P \vdash_u G$. In fact, the converse is true too [28], so that $P \vdash_o G$ iff $P \vdash_u G$. This, of course, doesn't mean that all proofs of the sequent $P \longrightarrow G$ are uniform, but that if there is a proof, then there is a uniform proof.

For example, the first proof in figure 1 is a not uniform proof, whereas the second one is.

In this way our interest in hereditary Harrop formulae is due to the fact that uniform proofs are complete with respect to intuitionistic logic for this class of formulae, rather then due to a desire to implement a particular style of theorem prover for intuitionistic logic. The notion of uniform proof is a stronger requirement than intuitionistic proof; for example, $\exists x p(x) \vdash_I \exists x p(x)$, but there is no uniform proof of the sequent $\exists x p(x) \longrightarrow \exists x p(x)$. In this way we are more interested in the strength of our conclusions than a particular proof system.

$$\dfrac{\dfrac{\dfrac{p(a), p(b) \longrightarrow p(a)}{p(a), p(b) \longrightarrow \exists x p(x)}\,\exists\text{-R}}{p(a) \wedge p(b) \longrightarrow \exists x p(x)}\,\wedge\text{-L} \qquad \dfrac{\dfrac{p(a), p(b) \longrightarrow p(a)}{p(a) \wedge p(b) \longrightarrow p(a)}\,\wedge\text{-L}}{p(a) \wedge p(b) \longrightarrow \exists x p(x)}\,\exists\text{-R}$$

Figure 1: Examples of uniform and non-uniform proofs

$$\dfrac{\dfrac{\dfrac{p(a) \longrightarrow p(a)}{p(a) \longrightarrow \exists x p(x)}\,\exists\text{-R} \quad q, p(a) \longrightarrow q}{p(a), \exists x p(x) \supset q \longrightarrow q}\,\supset\text{-L} \qquad \dfrac{\dfrac{p(b) \longrightarrow p(b)}{p(b) \longrightarrow \exists x p(x)}\,\exists\text{-R} \quad q, p(b) \longrightarrow q}{p(b), \exists x p(x) \supset q \longrightarrow q}\,\supset\text{-L}}{p(a) \vee p(b), \exists x p(x) \supset q \longrightarrow q}\,\vee\text{-L}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{p(y) \longrightarrow p(y) \qquad p(y), q \longrightarrow q}{p(y), p(y) \supset q \longrightarrow q}\,\supset\text{-L}}{p(y), \forall x(p(x) \supset q) \longrightarrow q}\,\forall\text{-L}}{\exists x p(x), \forall x(p(x) \supset q) \longrightarrow q}\,\exists\text{-L}$$

Figure 2: Uniform proofs involving indefinite formulae

3 Uniform Provability

Whilst the restriction to hereditary Harrop formulae is sufficient to guarantee the existence of uniform proofs, a natural question to ask is whether this restriction is necessary. As mentioned above, it was shown in [28] that if the antecedent is a set of definite formulae and the consequent a goal formula, then the sequent has a proof iff it has a uniform proof. The converse to this result is not (strictly) true. For example, the sequent $p(a) \vee p(b), (\exists x p(x) \supset q) \longrightarrow q$ has a uniform proof (given in Fig. 2), but the antecedent is not a set of definite formulae. Similarly, the sequent $\exists x p(x), \forall x(p(x) \supset q) \longrightarrow q$ has a uniform proof, but the antecedent is not a set of definite formulae.

Hence, it is not strictly true that for a uniform proof of $F_1 \longrightarrow F_2$ to exist we must have that F_1 is a definite formula. However it seems that the first uniform proof above relies on the fact that there is a uniform proof of $p(a), (\exists x p(x)) \supset q \longrightarrow q$ and of $p(b), (\exists x p(x)) \supset q \longrightarrow q$, in which both the antecedents are definite formulae. Similarly the second uniform proof above is dependent on the fact that the universally quantified variable may be replaced by any term, and hence the proof may be thought of as a template for a number of proofs of sequents of the form $p(t), \forall x(p(x) \supset q) \longrightarrow q$ for any term t. In this way there seems to be a more subtle relationship between uniform proofs and hereditary Harrop formulae. Indeed, as mentioned above, it is not possible to give a strict classification of the largest class of

$$\dfrac{\dfrac{p(a) \longrightarrow p(a)}{p(a) \longrightarrow p(a) \vee p(b)} \vee\text{-R} \quad \dfrac{p(b) \longrightarrow p(b)}{p(b) \longrightarrow p(a) \vee p(b)} \vee\text{-R}}{p(a) \vee p(b) \longrightarrow p(a) \vee p(b)} \vee\text{-L}$$

Figure 3: Non-uniform proof

formulae for which uniform proofs are complete, but it does seem that there is a relationship between uniform proofs and hereditary Harrop formulae which may be elucidated.

A result reported in [25] is that for sequents of the form $\Gamma \longrightarrow G$ where Γ is a set of definite formulae, there are no occurrences of the $\exists$-L or $\vee$-L rules. Hence, if there is an intuitionistic proof of a sequent in which antecedent is a definite formula and the consequent a goal formula, then there are no occurrences of the $\exists$-L and $\vee$-L rules and the sequent has a uniform proof. Thus we may conject that if a uniform proof of $F_1 \longrightarrow F_2$ contains no occurrences of either of these rules, then F_1 is a definite formula and F_2 is a goal formula. This again is not true, as there may be parts of the formula F_1 which ensure that F_1 is not a definite formula, but are not used in the proof. For example, the sequent $\exists x q(x), p(a) \longrightarrow p(a) \vee p(b)$ has a uniform proof, due to the fact that $p(a) \longrightarrow p(a) \vee p(b)$ has a uniform proof, and hence so does $F, p(a) \longrightarrow p(a) \vee p(b)$ for any formula F. This means that the relationship between a sequent $F_1 \longrightarrow F_2$ and some "equivalent" sequent $D \longrightarrow G$ will require more investigation. In particular, the role of the rules $\exists$-L and $\vee$-L need examination.

An interesting observation which may be made at this point is that the elimination rules for $\exists$ and $\vee$ in the natural deduction system for intuitionistic logic are criticised by Girard [10], because they destroy the subformula property of that system. This is due to the fact that there is a formula used in the proof which "has no structural link with the formula which is eliminated.". Whilst the sequent calculus presentation of intuitionistic logic does not have this property, it does involve creating "branch points" in the proof. For example, the instance of the $\vee$-L rule in the proof in Fig. 3 may be thought of as branching the search into two different sub-proofs in a way which cannot be reversed, in that we cannot interchange the order of the $\vee$-R and $\vee$-L rules.

In this way the $\vee$-L rule forces us to choose between possibilities, and hence this choice is reflected through the rest of the proof. The presence of this form of indefinite information is what prevents us from interchanging the order of the rules so that the subformula property may be used in the proof search process. In this way uniform proofs allow us to focus attention on the consequent, and only consider the antecedent when the consequent is an atom, whereas otherwise we may have to use a left rule at a seemingly arbitrary point in order to find a proof.

Apart from $\supset$-L, the left rules may be thought of as converting the antecedent into a desired form so that the appropriate right rules may be used. Hence, from a logic programmer's point of view it will often be useful to perform these manipulations before starting the "main" proof, as it were. This will be the case if we can interchange the order of the rules when a right rule precedes a left one. As noted above, the nature of the $\exists$-L and $\vee$-L rules

may make this difficult, and so it may not always be possible to re-arrange a given proof so that all the manipulation of the assertions can be done prior to the proof search process. However, there are some conditions under which this can be done.

For these reasons we introduce below the concept of a *definite* proof.

DEFINITION 3.1 *A proof* Ξ *is* definite *if* Ξ *contains no occurrences of either the* $\exists$*-L rule or the* $\vee$*-L rule.*

We denote definite provability by $\vdash_d$*.*

For this reason we will sometimes refer to the $\exists$-L and $\vee$-L rules as *indefinite* rules.

As mentioned above, it was shown in [25] that definite proofs are complete with respect to intuitionistic provability for a large fragment of hereditary Harrop formulae (although it is not difficult to extend that proof to full hereditary Harrop formulae). Below we state the generalisation of this result for hereditary Harrop formulae. The proof is straightforward, and is left to the reader.

PROPOSITION 3.2 *Let* Γ *be a set of definite formulae, and let* G *be a goal. Then any proof* Ξ *of* $\Gamma \longrightarrow G$ *is definite.*

Thus definite proofs are complete for definite formulae.

As Figure 2 shows, it is not true that $\Gamma \vdash_u F \Rightarrow \Gamma \vdash_d F$, but the converse is true, i.e. that $\Gamma \vdash_d F \Rightarrow \Gamma \vdash_u F$. In other words, a sequent with a definite proof has a uniform proof, but a uniform proof need not be definite.

In order to establish this result, we need to introduce some more terminology.

DEFINITION 3.3 *A proof is* RL *if there is a sequent in the proof which is the conclusion of a right rule and a premise of a left rule.*

A proof is LR *if it is not RL.*

A proof is locally LR *if the only occurrence of a sequent which is the conclusion of a right rule and a premise of a left rule is of the form*

$$\dfrac{\dfrac{\Xi_1}{\Gamma \longrightarrow B}\ ?\text{-}R \qquad \dfrac{\Xi_2}{\Gamma, C \longrightarrow F}}{\Gamma, B \supset C \longrightarrow F}\ \supset\text{-}L$$

where the sequent $\Gamma, C \longrightarrow F$ *is either initial or the conclusion of a left rule, and* Ξ_1 *and* Ξ_2 *are locally LR.*

We will often refer to the situation where a sequent is the conclusion of a right rule and a premise of a left rule as the right rule *immediately preceding* the left rule.

It should be clear that any proof is either RL or LR, and that any LR proof is a locally LR proof. Hence, if we can convert an RL proof into a locally LR proof under some conditions, then any sequent satisfying the conditions will have a proof iff it has a locally LR proof.

It also is not hard to see that a proof is locally LR iff it is uniform. This is shown in the proposition below, the proof of which is left to the reader.

PROPOSITION 3.4 *Let F be a negation-free formula, and let Γ be a set of negation-free formulae. Then a proof Ξ of $\Gamma \longrightarrow F$ is uniform iff Ξ is locally LR.*

Now we proceed to investigate the conditions under which we may transform an RL proof into a locally LR proof. Clearly this will follow if we can "commute" the rules, i.e. given an instance of a right rule immediately preceding a left rule, we can find a sub-proof starting from the same sequents in which the left rule precedes the right rule. We may then percolate this commutation as far as possible through the proof, finally arriving at a locally LR proof.

An exhaustive analysis of such *permutation properties* of the proof rules of intuitionistic logic was given by Kleene [19]. The lemma below, which is an instance of Lemma 7 of [19], summarises the permutation properties relevant to this paper. A similar permutation lemma has been used to advantage by Beeson [2].

LEMMA 3.5 (KLEENE'S PERMUTATION LEMMA) *If there is a proof of a given sequent in which there is a subproof in which either $\wedge$-R, $\vee$-R, $\exists$-R, $\forall$-R or $\supset$-R immediately precedes either $\wedge$-L, $\forall$-L or $\supset$-L on the right, then there is a proof of the same sequent with this subproof replaced by one in which the left rule precedes the right rule.*

The proof of this lemma is straightforward but somewhat involved; details may be found in [19] (a direct proof of the weaker result stated above may be found in [12]).

We now show how this lemma may be used to convert an RL proof which does not contain the indefinite connectives into a locally LR proof.

LEMMA 3.6 *Let F be a negation-free formula, and let Γ be a set of negation-free formulae. If $\Gamma \longrightarrow F$ has an RL proof which is definite, then $\Gamma \longrightarrow F$ has a locally LR proof.*

PROOF

Consider the occurrence of a right rule immediately preceding a left rule which is closest to the root of the proof tree. As there are no occurrences of $\exists$-L or $\vee$-L in the proof, the right rule must be one of $\wedge$-R, $\vee$-R, $\exists$-R, $\forall$-R or $\supset$-R followed by one of $\wedge$-L, $\forall$-L or $\supset$-L. Hence by lemma 3.5, we may interchange the order of the rules unless the right rule occurs on the left of the $\supset$-L rule.

In this way we may permute the right rule downwards until either we reach the root of the tree or the conclusion of the right rule is the left premise of the $\supset$-L rule or a premise of another right rule. Note that by our original choice of the occurrence of a right rule immediately preceding a left rule, the only rules that may follow this latter right rule are other right rules.

We may then perform this permutation for an occurrence of a right rule immediately preceding a left rule which is next closest to the root (i.e. is closer to the leaves than the original), and so on, until we have performed this operation for all occurrences of a right rule whose conclusion is a premise of a left rule. The only such occurrences that remain are those in which the conclusion of the right rule is the left premise of the $\supset$-L rule, and so the resulting proof is locally LR.

We are now in a position to show the result claimed earlier, i.e. that if a sequent has a definite proof, then it has a uniform proof.

THEOREM 3.7 *Let F be a negation-free formula, and let Γ be a set of negation-free formulae. Then*

$$\Gamma \vdash_d F \Rightarrow \Gamma \vdash_u F$$

PROOF Let Ξ be a definite proof of $\Gamma \longrightarrow F$. If Ξ is LR, then clearly Ξ is uniform. Otherwise, by lemma 3.6, there is a uniform proof Ξ' of $\Gamma \longrightarrow F$.

The above theorem may be thought of as showing that if we ignore the $\exists$-L and $\vee$-L rules, then we need only consider uniform proofs, i.e. uniform proofs are complete with respect to provability in intuitionistic logic provided $\exists$-L and $\vee$-L are not used.

Note also the strength of the contrapositive of the theorem, i.e. that if $\Gamma \longrightarrow F$ has a proof but no uniform proof, then all proofs of $\Gamma \longrightarrow F$ contain an occurrence of an indefinite rule.

4 Definiteness

We saw in the previous section how any definite proof may be converted into a uniform proof, and that for any provable sequent which does not have a uniform proof, all proofs of the sequent contain an occurrence of $\exists$-L or $\vee$-L. Thus an obvious way to ensure the completeness of uniform proofs is to restrict the class of formulae so that the indefinite rules become redundant.

One such class of formulae are definite formulae, and the redundance of the indefinite rules for definite formulae is precisely why definite formulae are interesting. Definite formulae seem very apt for constructive consequence, as they force the programmer to present his or her knowledge in a relatively strong way. We may think of an indefinite formula as conveying less information than a definite one. For example, the formula $\exists x p(x)$ carries less information than the formula $p(t)$, which may be used to derive the former one. Indeed, if we may imagine an intuitionistic programmer asserting that $\exists x p(x)$ is true, we may expect him to be able to construct a term t such that $p(t)$ is true. In fact this is a requirement if we insist upon constructive consequence, as $\exists x p(x) \vdash \exists x p(x)$, but we cannot derive the truth of any instance of $p(x)$. Hence we may imagine a compiler taking as input a set of formulae, and retaining only the definite parts of the formulae, as the indefinite parts do not provide us with enough information to make them useful.

In this way it seems that there is a strong connection between definite proofs and definite formulae, which is that given a definite proof of $\Gamma \longrightarrow F$, we may extract a set of definite formulae and a goal formula from the sequent, in the manner briefly described above. A more precise description is given below. We denote by $\top$ the formula "true".

DEFINITION 4.1 *Let F be a negation-free formula and A be an atom. Then we define*

$$\begin{aligned}
\mathrm{def}(A) &= A \\
\mathrm{def}(F_1 \wedge F_2) &= \mathrm{def}(F_1) \wedge \mathrm{def}(F_2) \\
\mathrm{def}(F_1 \vee F_2) &= \top \\
\mathrm{def}(\exists x F) &= \top
\end{aligned}$$

$$
\begin{aligned}
\mathrm{def}(\forall x F) &= \forall x\, \mathrm{def}(F) \\
\mathrm{def}(F_1 \supset F_2) &= \mathrm{goal}(F_1) \supset \mathrm{def}(F_2)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{goal}(A) &= A \\
\mathrm{goal}(F_1 \wedge F_2) &= \mathrm{goal}(F_1) \wedge \mathrm{goal}(F_2) \\
\mathrm{goal}(F_1 \vee F_2) &= \mathrm{goal}(F_1) \vee \mathrm{goal}(F_2) \\
\mathrm{goal}(\exists x F) &= \exists x\, \mathrm{goal}(F) \\
\mathrm{goal}(\forall x F) &= \forall x\, \mathrm{goal}(F) \\
\mathrm{goal}(F_1 \supset F_2) &= \mathrm{def}(F_1) \supset \mathrm{goal}(F_2)
\end{aligned}
$$

We also define $def(\{F_1, \ldots F_n\}) = \bigcup_{i=1}^{n} \{\mathrm{def}(F_i)\}$.

Note that $\mathrm{def}(F)$ is either $\top$ or a definite formula, and that $\mathrm{goal}(F)$ is a goal formula. Note also that $\mathrm{goal}(F)$ can never be $\top$.

We can now prove the following useful lemma.

LEMMA 4.2 *Let F be a negation-free formula.*
Then

1. $F \vdash_I \mathrm{def}(F)$
2. $\mathrm{goal}(F) \vdash_I F$

The proof of this lemma is a straightforward induction on the size of F.

In this way we see that $\mathrm{def}(F)$ and $\mathrm{goal}(F)$ preserve certain information, in that anything deducible from $\mathrm{def}(F)$ is deducible from F, and that anything deducible from F is deducible from $\mathrm{goal}(F)$. In addition, as shown below, the converse relationships hold for uniform provability.

PROPOSITION 4.3 *Let F be a negation-free formula, and let Γ be a set of negation-free formulae.*
If $\Gamma \vdash_d F$, then

1. $\mathrm{def}(\Gamma) \vdash_u F$
2. $\Gamma \vdash_u \mathrm{goal}(F)$
3. $\mathrm{def}(\Gamma) \vdash_u \mathrm{goal}(F)$

PROOF
By theorem 3.7, we have that $\Gamma \vdash_u F$.

We proceed by induction on the height of the (uniform) proof.

The base case occurs when F is just an atom A, and $A \in \Gamma$. Clearly $A \in \mathrm{def}(\Gamma)$, and so $\mathrm{def}(\Gamma) \vdash_u A$.

Hence the hypothesis is that the proposition holds for all proofs of no more than a given size.

Consider the final rule used in the proof. There are eight cases, of which we will only present the argument for two; the others are similar.

$\wedge$-R: In this case the previous sequents must be $\Gamma \longrightarrow F_1$ and $\Gamma \longrightarrow F_2$, and by the hypothesis we have that $\mathrm{def}(\Gamma) \vdash_u F_1$ and $\mathrm{def}(\Gamma) \vdash_u F_2$, i.e. $\mathrm{def}(\Gamma) \vdash_u F_1 \wedge F_2$.

Also by the hypothesis we have that $\Gamma \vdash_u \mathrm{goal}(F_1)$ and $\Gamma \vdash_u \mathrm{goal}(F_2)$, which is just $\Gamma \vdash_u \mathrm{goal}(F_1 \wedge F_2)$.

Again by the hypothesis we have that $\mathrm{def}(\Gamma) \vdash_u \mathrm{goal}(F_1)$ and $\mathrm{def}(\Gamma) \vdash_u \mathrm{goal}(F_2)$, which is just $\mathrm{def}(\Gamma) \vdash_u \mathrm{goal}(F_1 \wedge F_2)$.

$\supset$-L: In this case, as the proof is uniform, F must be just an atom A, and so as $\mathrm{goal}(A) = A$, we need only show 1.

The previous sequents must be $\Gamma \longrightarrow B$ and $\Gamma, C \longrightarrow A$, and by the hypothesis we have that both $\mathrm{def}(\Gamma) \vdash_u \mathrm{goal}(B)$ and $\mathrm{def}(\Gamma), \mathrm{def}(C) \vdash_u A$, and so $\mathrm{def}(\Gamma), \mathrm{goal}(B) \supset \mathrm{def}(C) \vdash_u A$, i.e. $\mathrm{def}(\Gamma, B \supset C) \vdash_u A$.

COROLLARY 4.4 *Let F be a negation-free formula, and let Γ be a set of negation-free formulae.*

If $\Gamma \vdash_d F$, then there is a set of definite formulae Γ' and a goal formula G such that

1. $\Gamma \vdash_I \bigwedge \Gamma'$
2. $G \vdash_I F$
3. $\Gamma' \vdash_u G$

PROOF Follows immediately from propositions 4.2 and 4.3.

In this way we see that if $\Gamma \longrightarrow F$ has a definite proof, then not only does the same sequent have a uniform proof, but also we may extract a set of definite formulae Γ' from Γ such that $\Gamma \vdash_I \bigwedge \Gamma'$ and $\Gamma' \vdash_u F$, and a goal formula G from F such that $G \vdash_I F$ and $\Gamma \vdash_u G$. In this way we may think of this result as a version of Craig's Interpolation theorem [3], in that given a proof of $\Gamma \longrightarrow F$, then provided that there are no occurrences of $\exists$-L or $\vee$-L in the proof, then we can interpolate a definite formula D such that $\Gamma \vdash_I D$ and $D \vdash_I F$. Thus given Γ, we can derive a definite formula which is provable from Γ and has the same consequences, provided that we consider only definite proofs. Hence, definite formulae arise naturally out of consideration of definite proofs, which in turn arise naturally out of consideration of the permutability of the left and right rules in intuitionistic logic.

In the next section, we examine a further development of the extraction of definite formulae from proofs.

5 Maximality of Information and Definiteness

The result above may be interpreted as showing what efficiencies we can make in the process of searching for a proof provided that we restrict our attention to definite proofs. As described above, we may think of this in a similar manner to the Interpolation theorem. A criticism which may be made of this approach is that whilst indefinite formulae may contain less information than definite ones, the above transformation of indefinite formulae to definite

formulae actually loses some information, as the indefinite parts of the formulae are ignored. Also, the requirement that the proof be definite is a stronger one than merely requiring the proof to be uniform. Hence it may be interesting to examine what may be done to preserve (or strengthen) the original information rather than weakening it, and to see if uniform proofs are still sufficient in these circumstances.

An obvious alternative approach to extracting definite information from a proof is to find a definite formula of which the premise is a consequence, rather than a definite formula which is a consequence of the premise. For example, in the first proof in Figure 2, it is clear that there is a uniform proof of the sequent $p(a), \exists x p(x) \supset q \longrightarrow q$, and that

$$p(a), \exists x p(x) \supset q \vdash_I (p(a) \vee p(b)) \wedge \exists x p(x) \supset q.$$

We may think of this approach as attempting to supply sufficient information in order to make the formula definite, rather than ignoring the indefinite parts of the formula, and hence we will be suggesting hypotheses which will make the formula true. This leads us to the concept of a *definite condition* and a *definite consequence*.

DEFINITION 5.1 *A* definite condition *of a formula F is a formula which is the same as F except that*

1. *Every positive occurrence of a subformula $\exists x F'$ in F is replaced by $F'[t/x]$ for some term t in which all variables of t are universally quantified in F and the scope of all such quantifiers includes $\exists x F'$.*

2. *Every positive occurrence of a subformula $F_1 \vee F_2$ in F is replaced by one of F_i, $i = 1, 2$*

We denote by $defprem(F)$ the set of all definite conditions of F.

If $\Gamma = \{F_1, \ldots F_n\}$ is a set of formulae, then D is a definite condition of Γ if D is of the form $D_1 \wedge \ldots \wedge D_n$ where D_i is a definite condition of F_i.

Note that the first condition ensures that any free variables in a definite condition of a formula F are also free in F itself.

DEFINITION 5.2 *A* definite consequence *of a formula F is a formula which is the same as F except that*

1. *Every negative occurrence of a subformula $\exists x F'$ in F is replaced by $F'[t/x]$ for some term t in which all variables of t are universally quantified in F and the scope of all such quantifiers includes $\exists x F'$.*

2. *Every negative occurrence of a subformula $F_1 \vee F_2$ in F is replaced by one of F_i, $i = 1, 2$*

We denote by $defconc(F)$ the set of all definite consequences of F.

If $\Gamma = \{F_1, \ldots F_n\}$ is a set of formulae, then G is a definite consequence of Γ if G is of the form $G_1 \wedge \ldots \wedge G_n$ where G_i is a definite consequence of F_i.

Note again that a free variable appearing in a definite consequence is also a free variable in the original formula.

Note also that a definite condition of $\exists x F$ cannot contain any occurrence of x, and hence must produce a "ground witness" for x. For example, the only definite conditions of $\exists x\ p(x)$ are atoms of the form $p(t)$ where t is a ground term.

For existentially quantified variables appearing within the scope of a universally quantified variable, we may use the universally quantified variable to construct the witness. For example, one of the definite conditions of $\forall x \exists y\ p(x, y)$ is $\forall x\ p(x, f(x))$.

It should be clear that for negation-free formulae, definite conditions and definite consequences are definite and goal formulae respectively. A formal statement of this is given below.

PROPOSITION 5.3 *Let F be a negation-free formula. Then*

1. *Any definite condition of F is a definite formula.*
2. *Any definite consequence of F is a goal formula.*

It should be clear that if D is a definite condition of F_1 and G is a definite consequence of F_2, then $G \supset D$ is a definite condition of $F_2 \supset F_1$, and $D \supset G$ is a definite consequence of $F_1 \supset F_2$.

Firstly we show a useful lemma.

LEMMA 5.4 *Let $\exists x F_1$ and F_2 be negation-free formulae, and let Γ be a set of negation-free formulae such that x does not occur free in Γ or F_2.*

Let Ξ be a proof of $\Gamma, F_1[y/x] \longrightarrow F_2$ where y does not occur free in F_1, F_2 or Γ.

If Ξ' is the result of replacing every occurrence of y in Ξ by a ground term t, then Ξ' is a proof of $\Gamma, F_1[t/x] \longrightarrow F_2$.

Note that we may rename the variables in Γ and F_2 so that x does not occur anywhere, free or bound, and so we may satisfy the condition on x for any $\exists x F_1$, F_2 and Γ.

Note that as y does not occur free in F_1, F_2 or Γ, variable capture in $F_1[t/x]$ is avoided. The proof of the lemma is a straightforward induction on the height of the proof, and hence is omitted.

Next we show that definite conditions and definite consequences behave in the expected manner.

PROPOSITION 5.5 *Let F be a negation-free formula. Then for any definite condition D of F and definite consequence G of F*

1. $D \vdash_I F$
2. $F \vdash_I G$

The proof is straightforward, and similar to that of the Substitutivity of Implication [1], and hence is omitted.

We may think of this as stating that D has more explicit information than F, so that if we were to consider an ordering of formulae in which $F_1 \leq F_2$ iff $F_2 \vdash_I F_1$, then the above proposition ensures that for any F, there is always a definite formula D such that $F \leq D$. Similar remarks apply to G, in that there is always a G such that $G \leq F$. In this way if we think of a lattice of formulae in which the partial order is (intuitionistic) provability, then any chain has a least upper bound which is a definite formula, and a greatest lower bound which is a goal formula. Thus we extrapolate from the formula to a more definite statement.

Next we show that definite conditions and definite consequences preserve uniform provability.

PROPOSITION 5.6 *Let F, F' be negation-free formulae, and let Γ be a set of negation-free formulae. Then for any definite condition D of F and any definite consequence G of F*

1. $\Gamma, F \vdash_u F' \Rightarrow \Gamma, D \vdash_u F'$
2. $\Gamma \vdash_u F \Rightarrow \Gamma \vdash_u G$

PROOF

We proceed by induction on the height of the uniform proof.

1. In the base case, F' is an atom A, in which case we have $\Gamma, F \vdash_u A$ and $A \in \Gamma \cup \{F\}$, so that either $A \in \Gamma$ and hence $\Gamma, D \vdash_u A$, or $A = F$, and as the only definite condition of A is A, it is clear that $\Gamma, D \vdash_u A$.

2. In the base case, F is an atom A such that $A \in \Gamma$, and as the only definite consequence of an atom A is A, it is clear that $\Gamma \vdash_u G$.

Hence we assume that the proposition is true for all uniform proofs of no more than a given size.

There are six cases, and again many of them are repetitive. Hence we only present the argument for selected cases.

leftrules: In this case the succedent must be just an atom A, and so the sequent is $\Gamma, F \longrightarrow A$.

1. Consider the rule used to derive the sequent $\Gamma, F \longrightarrow A$. By the hypothesis we may replace F by D anywhere in the previous sequent or sequents, and so the only remaining case is when F does not appear "intact" in the previous sequent/s. Hence we only need to consider applications of one of the left rules to F. Of these, only the quantifier cases are given below (the others are straightforward):

   $\exists$-L: F must be of the form $\exists x F''$, and the previous sequent must be $\Gamma, F''[y/x] \longrightarrow A$ where y does not occur free in A, Γ or F, and hence by lemma 5.4 $\Gamma, F''[t/x] \vdash_u A$ for some ground term t, and by the hypothesis $\Gamma, D \vdash_u A$ where D is a definite condition of $F''[t/x]$, and hence of $\exists x F''$.

$\forall$-L: F must be of the form $\forall x F$, and the previous sequent must be $\Gamma, F[t/x] \longrightarrow A$, and hence by the hypothesis we have that $\Gamma, D \vdash_u A$, where D is a definite condition of $F[t/x]$. Now if x doesn't occur in D, then D is a definite condition of $\forall x F$ and we are done. Otherwise, $\forall x D$ is a definite condition of $\forall x F$, and by another application of $\forall$-L we have that $\Gamma, \forall x D \vdash_u A$.

2. As the only definite consequence of an atom A is A itself, it is clear that $\Gamma \vdash_u G$ where G is any definite consequence of A.

$\exists$-R:

1. The previous sequent must be $\Gamma, F \longrightarrow F'[t/x]$ for some $t \in \mathcal{U}'$, and by the hypothesis, if D is a definite condition of F then $\Gamma, D \vdash_u F'[t/x]$, i.e. $\Gamma, D \vdash_u \exists x F'$.

2. The previous sequent must be $\Gamma \longrightarrow F[t/x]$, and by the hypothesis, if G' is a definite consequence of $F[t/x]$, then $\Gamma \vdash_u G'$. Let G be any definite consequence of F. Now we know that either $G[t/x]$ or G is a definite consequence of $F[t/x]$. Now if x doesn't occur in G then G is then a definite consequence of $F[t/x]$ and of $\exists x F$, and so we have $\Gamma \vdash_u G$. Otherwise, $G[t/x]$ and $\exists x G$ are definite consequences of $F[t/x]$ and $\exists x F$ respectively, and so we have $\Gamma \vdash_u G[t/x]$, i.e. $\Gamma \vdash_u \exists x G$.

$\forall$-R:

1. The previous sequent must be $\Gamma, F \longrightarrow F'[y/x]$ where y is not free in Γ, F or F', and by the hypothesis, if D is a definite condition of F then $\Gamma, D \vdash_u F'[y/x]$, i.e. $\Gamma, D \vdash_u \forall x F'$.

2. The previous sequent must be $\Gamma \longrightarrow F[y/x]$ where y is not free in Γ and F, and by the hypothesis if G' is a definite consequence of $F[y/x]$, $\Gamma \vdash_u G'$. Let G be any definite consequence of F. Now we know that either $G[y/x]$ or G is a definite consequence of $F[y/x]$. Now if x doesn't occur in G then G is then a definite consequence of $F[y/x]$ and of $\forall x F$, and so we have $\Gamma \vdash_u G$. Otherwise, $G[y/x]$ and $\forall x G$ are definite consequences of $F[y/x]$ and $\forall x F$ respectively, and so we have $\Gamma \vdash_u G[y/x]$, i.e. $\Gamma \vdash_u \forall x G$.

From this result we may prove the following corollary.

COROLLARY 5.7 *Let F be a negation-free formula, and let Γ be a non-empty finite set of negation-free formulae.*

If $\Gamma \vdash_u F$, then for any definite condition D of Γ, $D \vdash_u F$.

PROOF

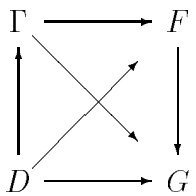
We proceed by induction on the size of Γ. When $\Gamma = \{F'\}$, i.e. Γ is just a singleton set, the result follows immediately from proposition 5.6.

Hence we assume that the corollary holds for all non-empty sets of formulae of no more than a given size. Let $\Gamma = \Gamma' \cup \{F'\}$.

If $\Gamma', F' \vdash_u F$, then by proposition 5.6 we have $\Gamma', D \vdash_u F$ for any definite condition D of F', and by the hypothesis we have $D', D \vdash_u F$ for any definite condition D' of Γ', i.e. $D'' \vdash_u F$ for any definite condition D'' of Γ.

We may think of the above proposition and corollary as a form of "extrapolation" result, in that given a uniform proof of $\Gamma \longrightarrow F$ we can find a definite formula D and a goal formula G such that $D \vdash_I \bigwedge \Gamma$, $D \vdash_u F$, $F \vdash_I G$ and $\Gamma \vdash_u G$, and as a consequence, $D \vdash_u G$.

This relationship may be represented by the diagram below, where the vertical arrows denote intuitionistic provability, and the horizontal and diagonal arrows denote uniform provability.



Thus given any uniform proof, we can find a definite formula D and a goal formula G which preserve the appropriate provability relationships. Hence we may conclude that this result supports our contention that hereditary Harrop formulae are maximal with respect to uniform proofs, in that any sequent which has a uniform proof may be thought of as an incomplete specification of a hereditary Harrop sequent $D \longrightarrow G$.

6 Discussion

A point to note about uniform proofs is that there is no specification of how atomic goals are to be handled. However, it is clear that there is no great mystery about this process; in particular, the $\forall$-L, $\wedge$-L and $\supset$-L rules are used in a certain way to derive the next goal from an atomic goal — for example using Miller's *backchain* rule [25]. Hence it seems reasonable to consider such a process as the uniform rule for an atom, and in this way strengthen the notion of uniform proof to include this case. It should be noted that a related issue is the notion of a *simple proof*, due to Miller [25]; a proof is simple if all occurrences of the $\supset$-L rule in the proof are such that the right hand premise is initial. This essentially makes the $\supset$-L rule into a unary one, and seems to point the way towards the backchain rule.

Thus by combining the $\forall$-L, $\wedge$-L and $\supset$-L rules in a certain way, we can derive a stronger notion of uniform proof. In fact, this is precisely what is given by the definition of $\vdash_o$, and, as mentioned above, it is shown in [25] that $\Gamma \vdash_u G$ iff $\Gamma \vdash_o G$. Hence we may think of $\vdash_o$ as providing a stronger notion of uniformity than uniform proofs alone.

However, we may consider the existence of this stronger notion of uniformity as evidence that perhaps the notion of uniform proof itself not quite what is needed for our purposes; it would seem that a stronger notion is needed in order to determine logic programming languages. In particular, it seems that a specification of how atoms are to be dealt with is required. This may be interpreted as evidence that whilst the completeness of uniform provability seems to be necessary for logic programming languages, a stronger criterion is needed for analyses akin to the one in this paper.

7 Conclusions and Further Work

We have seen how restricting first-order intuitionistic proofs in certain ways leads to some results which ensure that the task of searching for a proof is made more feasible than in the general case. We may think of the restrictions as ensuring that the information contained in the formulae is presented in a maximal way, so that we do not need to waste time discovering this information during the computation process. This may be thought of as requiring that we only consider proofs in a "normal form".

One way to think of this maximal class is to consider it as a "definite" or "uniform" sub-logic of intuitionistic logic, with a more restricted notion of provability. As noted in section 1, intuitionistic provability alone does not guarantee constructive consequence, i.e. that the following hold for any (possibly empty) set of formulae Γ:

$$\Gamma \vdash \exists x F \Leftrightarrow \Gamma \vdash F[t/x]$$
$$\Gamma \vdash F_1 \vee F_2 \Leftrightarrow \Gamma \vdash F_1 \text{ or } \Gamma \vdash F_2$$

However the following equivalences do hold in intuitionistic logic:

$$\Gamma \vdash \forall x F \Leftrightarrow \Gamma \vdash F[y/x]$$
$$\Gamma \vdash F_1 \wedge F_2 \Leftrightarrow \Gamma \vdash F_1 \text{ and } \Gamma \vdash F_2$$
$$\Gamma \vdash F_1 \supset F_2 \Leftrightarrow \Gamma, F_1 \vdash F_2$$

where y is not free in Γ or F.

Hence we see that imposing constructive consequence on intuitionistic logic gives us precisely goal-directed provability. Alternatively, imposing goal-directed provability on intuitionistic logic gives us constructive consequence. Thus we may think of hereditary Harrop formulae as an important fragment of intuitionistic logic, in that they seem to be the largest class of formulae for which the notion of constructive consequence, and hence goal-directed provability, can be guaranteed. [2]

We have also seen some relationship between the restricted classes of proofs and formulae and the more general classes, and in particular how definite formulae and goals may be extracted from an arbitrary uniform proof, and that the extracted formulae preserve uniform provability. It is possible that this result may be useful for program specification, in that if a specification is given as a first-order formula (without negation), then the extraction process described above may be thought of as finding a definite formula (i.e. a program) which satisfies the specification.

Another possible development is to investigate a slightly different form of universal quantification than that given above. Given a Herbrand Universe $\mathcal{U}$, if $p(t)$ holds for all $t \in \mathcal{U}$, it seems reasonable to conclude that $\forall x p(x)$ holds. We may think of this as an extensional approach to universal quantification, in that we need only establish the truth of all instances of the goal is true, whereas in the case discussed above, we need to establish the truth of some fixed but arbitrarily chosen term. For example, given the following program, $p(t)$ succeeds for every $t \in \{a, f(a), f(f(a)), \ldots\}$, but $p(y)$ fails. [3].

[2] In fact, the natural logic in which to interpret hereditary Harrop formulae is slightly stronger than intuitionistic logic; see [11, 13] for details.

[3] Note, however, that the goal $\exists x p(x)$ succeeds

$$p(a)$$
$$\forall x\ p(f(x))p(x)$$

Hence we may wish for a slightly stronger rule for universal quantification, in that if $P \vdash_u p(y)$, then $P \vdash_u p(t)$ for any ground term t. As discussed in [13], this will require some form of structural induction. This difference between the two rules is similar to the relationship between D and $|D|$. It is clear that $D \vdash_I D'$ for any $D' \in |D|$, and so (assuming we may write infinitary formulae) $D \vdash_I \bigwedge |D|$, but the converse is not true.

We may think of this rule as making an assumption about the way that terms are generated. In particular, we need to know that no new terms will be generated during the computational process. This may be thought of as a consequence of our use of a universe of terms, i.e. the Herbrand universe, and that this set of terms is fixed prior to computation.

We may think of this approach to terms as intermediate between those of classical logic and intuitionistic logic. In intuitionistic logic, new objects may be constructed during mathematical investigation, and so one cannot think of the set of terms as complete. Indeed, a new term may be constructed during computation. In the classical case, we do not think in terms of constructions, but that the terms have been delineated beforehand, even if we do not know what all the terms are. Hence, whilst we cannot necessarily list all terms, we know that the terms cannot be added to during computation. The view that seems to make the most sense for logic programming is that new terms may be generated at a later time, but this generation cannot take place during the computation process. In this way at any given time we do not necessarily know that no more terms will be added, but we do know that these terms do not arise as a result of our computations (i.e. they are added by the programmer later). This philosophy seems to fit in well with many-sorted logic, in that one can reach conclusions such as for all terms t of a given type τ, $p(t)$ holds, and hence $\forall x : \tau\, p(x)$. Thus whilst we may later discover that $p(t')$ holds for some term t' which is not of type τ, we can still reach a quantified conclusion which will not be invalidated at a later time.

In this way an infinitary intermediate logic may be useful for further proof-theoretic investigations along these lines. We will take up this theme in a later paper.

8 Acknowledgements

My thanks go to Dale Miller for many interesting discussions and weighty deliberations. Discussions with David Pym have also been enlightening, and comments from some anonymous referees were very helpful.

References

[1] P. Andrews, *An Introduction to Mathematical Logic and Type Theory*, Academic Press, 1986.

[2] M. Beeson, Some Applications of Gentzen's Proof Theory in Automated Deduction, *Extensions of Logic Programming: International Workshop, Tübingen FRG, December*

1989, P. Schroeder-Heister (ed.), Lecture Notes In Artificial Intelligence 475, 101-156, Springer-Verlag, 1991.

[3] G.S. Boolos and R.C. Jeffrey, *Computability and Logic*, Cambridge University Press, 1980.

[4] W.F. Clocksin and C.S. Mellish, *Programming in Prolog*, Springer-Verlag, 1984.

[5] R.L. Constable and S.F. Smith, Computational Foundations of Basic Recursive Function Theory, *Proceedings of the Symposium on Logic in Computer Science* 360-371, Edinburgh, July, 1988.

[6] H.B. Curry, The Permutability of Rules in the Classical Inferential Calculus, *Journal of Symbolic Logic* 17, 245-8, 1952.

[7] D. van Dalen, Intuitionistic Logic, in *Handbook of Philosophical Logic Volume III: Alteratives to Classical Logic* 225-339, D.M. Gabbay and F. Guenthner (eds.), Reidel, 1986.

[8] D. Gabbay, N-Prolog: An Extension of Prolog with Hypothetical Implication.II. Logical Foundations and Negation as Failure, *Journal of Logic Programming* 2:251-283, 1985.

[9] D. Gabbay and U. Reyle, N-Prolog: An Extension of Prolog with Hypothetical Implications. I., *Journal of Logic Programming* 1:319-355, 1984.

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

[11] J. Harland, On Normal Forms and Equivalence for Logic Programs, *Proceedings of the Joint International Conference and Symposium on Logic Programming* 146-160, Washington D.C, November, 1992.

[12] J. Harland, A Proof-Theoretic Analysis of Logic Programming, Technical Report 90/21, Department of Computer Science, University of Melbourne, 1990.

[13] J. Harland, *On Hereditary Harrop Formulae as a Basis for Logic Programming*, Ph.D. Thesis, Department of Computer Science, University of Edinburgh, July, 1991.

[14] J. Harland and D. Pym, The Uniform Proof-theoretic Foundation of Linear Logic Programming, *Proceedings of the International Logic Programming Symposium* 304-318, San Diego, October, 1991.

[15] J. Harland and D. Pym, The Uniform Proof-theoretic Foundation of Linear Logic Programming, Report ECS-LFCS-90-124. University of Edinburgh, 1990. Also published as Technical Report 90/26, Department of Computer Science, University of Melbourne.

[16] R. Harper, F. Honsell and G. Plotkin, A Framework for Defining Logics, *Proceedings of the Symposium on Logic in Computer Science*, Ithaca, June, 1987.

[17] J. Hodas and D. Miller, Logic Programming in a Fragment of Intuitonistic Linear Logic, *Proceedings of the Symposium on Logic in Computer Science* 32-42, Amsterdam, July, 1991.

[18] S.C. Kleene, *Introduction to Metamathematics*, North-Holland, 1952.

[19] S.C. Kleene, Permutability of Inferences in Gentzen's Calculi LK and LJ, *Memoirs of the American Mathematical Society* 10, 1952.

[20] J.W. Lloyd, *Foundations of Logic Programming*, Springer-Verlag, Berlin, 1984.

[21] D.C. McCarty, Intuitionism: An Introduction to a Seminar, *Journal of Philosophical Logic* 12:105-149, 1983.

[22] L.T. McCarty, Clausal Intuitionistic Logic I. Fixed Point Semantics, *Journal of Logic Programming* 5:1:1-32, 1988.

[23] L.T. McCarty, Clausal Intuitionistic Logic II. Tableau Proof Procedures, *Journal of Logic Programming* 5:2:93-132, 1988.

[24] P. Martin-Löf, Constructive Mathematics and Computer Programming, in *Logic, Methodology and Philosophy of Science VI*, L.J. Cohen, J. Los, H. Pfeiffer and K.D. Podewski (eds.), North-Holland, 1982.

[25] D.A. Miller, A Logical Analysis of Modules in Logic Programming, *Journal of Logic Programming* 6:79-108, 1989.

[26] D.A. Miller, Lexical Scoping as Universal Quantification, *Proceedings of the International Conference on Logic Programming* 268-283, Lisbon, June, 1989.

[27] D.A. Miller and G. Nadathur, Some Uses of Higher-Order Logic in Computational Linguistics, *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics* 247-255, 1986.

[28] D.A. Miller, G. Nadathur, F. Pfenning and A. Scedrov, Uniform Proofs as a Foundation for Logic Programming, *Annals of Pure and Applied Logic* 51:125-157, 1991.

[29] G. Nadathur and D.A. Miller, Higher-Order Horn Clauses *Journal of the Association for Computing Machinery* 37:4:777-814, ctober, 1990.

[30] G. Nadathur and D.A. Miller, An Overview of λProlog, *Proceedings of the International Conference and Symposium on Logic Programming*, Seattle, August, 1988.

[31] P. Schroeder-Heister, Hypothetical Reasoning and Definitional Reflection in Logic Programming, *Extensions of Logic Programming: International Workshop, Tübingen FRG, December 1989*, P. Schroeder-Heister (ed.), Lecture Notes In Artificial Intelligence 475, 327-340, Springer-Verlag, 1991.

[32] L. Sterling and E. Shapiro, *The Art of Prolog*, MIT Press, 1986.

[33] S.-A. Tärnlund, Horn Clause Computability, *BIT* 17:215-226, 1977.