ADVANCED THEOREM-PROVING TECHNIQUES FOR RELEVANT LOGICS

P.B. THISTLEWAITE, R.K. MEYER and M.A. MCROBBIE

Abstract

In this paper we discuss some of the history of and motivations for non-classical theorem-proving, concentrating on the uses within logic and AI of automated theorem-proving (ATP) systems based on relevant logic. We discuss various techniques which have been incorporated in the program KRIPKE of [37] and [18] for deciding theoremhood in a range of relevant logics.

1. Logic and Relevance

Morgan in [25] notes the importance for artificial intelligence which is claimed for nonclassical logics by various authors, and goes on to cite a considerable amount of literature to justify this claim. One of the central uses of modal logic has been to explicate the semantics of programming languages. Other non-classical logics, including 'fuzzy', temporal and relevant logics, have had their proponents, with the particular application varying with the logic in question. Regarding relevant logics, [2] has suggested their use in question-answer systems, and [36] has discussed their use in deductive database systems.

Despite the apparent and growing interest in non-classical logic, not every researcher into methods of automated theorem-proving (ATP) has been sympathetic to the use or development of non-classical ATP systems based explicitly on non-classical proof procedures. This resistance to non-classical ATP would seem to have two strands to it: some (e.g. [6]) intimate a rejection of non-classical logic itself, and others (e.g. [40] [41]) who seem to see the interest or importance of non-classical notions, nonetheless believe that non-classical ATP does not require the use of specialized non-classical proof procedures. We disagree with each of these views, especially in the case of relevant logic.

Relevant logic challenges the classical account of what *counts* as a deduction, valid inference, or proof, and thus the matter is of central importance, for *proving* is the very object of the exercise. The claim is that classical logic gets the notion of implication wrong, and the corresponding ramification for classically-baded ATP is that its methods may be faulty and unreliable. Indeed, modal, intuitionistic and relevant logics have all at one time or another been motivated as presenting accounts of implication superior to the account given by classical logic. For example, the *strict* implication of the modal systems of [15] were motivated by Lewis as presenting a better account of implication, in that strict implication avoided several of the paradoxical properties of the classical, *material* implication since the following theorems of classical logic are not theorems of these systems where implication is cashed out as strict implication (or for that matter, of relevant logics)

(i) $p \rightarrow (q \rightarrow p)$
(ii) $\sim p \rightarrow (p \rightarrow q)$
(iii) $(p \rightarrow q) \vee (q \rightarrow p)$

It was pointed out that the following paradoxes of material implication (amongst others) remained theorems of the Lewis systems when cashed out in terms of strict implication

(iv) $(p \& \sim p) \rightarrow q$
(v) $p \rightarrow (q \vee \sim q)$

(It should also be noted that (i), (ii) and (iv) are theorems of intuitionistic logic as well.) Partly because of these paradoxes of strict implication, few of the logicians who were interested in investigating modal logics motivated this interest in terms of avoiding paradox, and indeed, Lewis himself abandoned this motivation too. Age has a way of betraying one's weaknesses, however, and so it is not surprising that the paradoxical theses, (i)-(v), concerning implication eventually came under renewed attack. This attack occurred in the 50's with the advent of the relevant family of logics. Relevant (or relevance) logics endeavour to confront the issues of implication including the paradoxes of both material and strict implication head on, and indeed none of (i)-(v) are theorems of any of the central relevant logics.

Broadly speaking, relevant logic regards a vertebrate theory of implication or valid *reasoning* as the cornerstone of a general theory of logic. This view has wide support within the logical community. For example, Quine claimed that

> "The chief importance of logic lies in implication..." [33] (p.xvi)

Quine, of course, believes that implication is properly understood solely in terms of truth-functional relations. Preservation of truth may be one aspect of good inferences, but having a diet consisting solely of truth-preservation makes for a poor theory of implication, as the paradoxes amply show. Relevant logics insist that the premisses of a valid implication be somehow *relevant* to what is implied. [1] and [35] present formal criteria for 'relevance', and we direct the reader to these texts for details of the relevant account of implication, and other philosophical and logical motivations for investigating relevant logics. Of course, logicians of the classical persuasion have tried to defend classical logic in the face of these criticisms by the proponents of relevant logic. [35] contains a good though perhaps partisan account of these defences. ([1])

One of the central observations of relevant logic is that while the deductive process cannot be held responsible for misreported facts, it would be disconcerting if the presence of inconsistency were to

([1]) It should be noted that relevant logics need not be seen as antagonistic to classical perspectives, despite the fact that some of the preferred motivations for them are openly so. Classical logic, which following [1] we call **TV**, and the principal relevant logic **R** can each be formulated with the same set of primitive connectives – for negation, conjunction, disjunction and implication – and a Hilbert-style axiomatization given such that the axioms and rules for **R** are a proper subset of those given for **TV** (see the next section for details). Taking **TV** and **R** to be thus talking about implication univocally, we see that the theorems of **R** are a proper subset of the theorems of **TV**, and so **R** rejects some classical theorems (like the paradoxes of implication). Alternatively, as Meyer in [20] argues, classical logic can be seen as being contained in **R**, and **R** can be viewed as an extension of classical logic. As is well-known, **TV**) can be axiomatized simply by taking connectives for negation, and conjunction and/or disjunction, as primitive (it is this that permits conjunctive normal forms and other clausal equivalences of classical logic that in turn permit resolution techniques). Formulated this way, **TV** is either seen as foregoing implicational formulas, or alternatively as defining an additional, material implication in the familiar way using negation and disjunction.

disable the deductive process itself. This does not happen in relevant logics. On the other hand, even if a classical ATP system resists *explicit* appeals to the paradoxes of material implication in order to prove some theorem, the paradoxes are nonetheless theorems of classical logic, and may reside in the classical deductive closure of a set of sentences under classical modus ponens. It was essentially considerations like these that encouraged Belnap in [2] and [3] to propose the use of relevant logics in deductive database systems and question-answer systems. Deductions over database information can in general be quite unlike classical deductions. For example, one of the defences of the classical account of good argumentation is that a sound argument is valid material implication plus the truth of premisses', but as Belnap rightly remarks especially of machine reasoning, we are often *compelled* by circumstance to reason with collected information that we've been *told* is true but which, whether we are aware of it or not, taken together is in fact inconsistent.

Handling inconsistent complex databases is just one of the areas in deductive database theory where employing relevant logics might help. Another problem concerning database management is, as Plaisted in [30] notes

> "To solve problems in the presence of large knowledge bases, it is important to be able to decide which knowledge is relevant to the problem at hand." (p. 79)

Indeed, Plaisted proposes several criteria for determining when pieces of information are relevant to the (possible) derivation of some conclusion, the main one being in terms of whether the literals (or propositional variables and constants) featuring in the information are *fully matched* (see p. 79). This 'literal matching' criterion of Plaisted's is directly related to the *variable sharing* criterion that Anderson and Belnap in [1] propose for relevantly valid deductions. Given these close connections, and the fact that [1] contains a wealth of formal and philosophical ideas regarding relevance, especially the relevance of premisses to conclusions, the utility of using them to partition large databases into (deductively) relevant parts is strongly indicated.

Granting then that relevant logics are important, and of interest to AI, it is disconcerting that little research has been conducted into developing efficient proof procedures for them. Part of the reason for this is that as Morgan rightly notes, many if not all non-classical (propositional) logics can be treated as higher-order (first-order) *theories* within classical logic. Indeed, some of the early research into non-classical ATP, such as that of [13], employed exactly this approach. However Morgan (p. 852) conjectured that, "If automated theorem proving is to be developed as a practical tool either for the logician in the area of non-classical logics or for the AI researcher wishing to use non-classical logics, it seems that avoidance of higher order techniques is desirable." Weyhrauch in [40] and Bibel in [6] explicitly or implicitly disagree. Yet it is never contested that second-order ATP, for example, is considerably more difficult than first-order ATP, and conversely that specialized systems incorporating a great deal of local knowledge about the problem domain will often perform considerably better on problems in that domain than a general purpose problem solver. So if the logic of a given situation suggests or demands that, say, modal notions feature centrally in some analysis of that situation, then it would seem foolish not to have and to use the most suitable tool for providing that analysis. To suggest otherwise would be akin to recommending a four-wheel drive vehicle to a competitor in the Grand Prix on the grounds that a Jeep can take you anywhere that a Formula One can.

In response to the general-purpose versus special-purpose line of criticism by Wos in [41], the matter is clearly an empirical one and will often be decided one way or the other depending on *how* general or special the problem to be solved is. Our contribution to this debate will be to present several problems from logic that can be solved using KRIPKE. We argue that these problems will be extremely difficult for extant general purpose systems to solve.

2. *Basic ATP Techniques for Relevant Logics*

Morgan in [25] outlines ATP methods for a range of non-classical logics including modal, many-valued, relevant and intuitionistic logics, but admits that these methods are not particularly suited to non-classical logics of the relevant or intuitionistic persuasion which, unlike modal logic, adopt non-classical notions of deduction. It is also important to note that the difficulties involved at various levels of the classical ATP program set in sooner relevantly; for example, Urquhart in [38] has recently proved that the principal relevant *propositional* logic, **R**, is undecidable, whereas undecidability does not arise classically until the first-order level. Inasmuch as first-order ideas are extensions of propositional ideas, a relevant ATP program requires a much more thorough understanding of relevant propositional ATP before embarking on ATP for higher-order relevant logics.

To this end, we will investigate the relevant logic **LR**.(2) Our propositional variables will be $\{p, q, r, p_1, \ldots\}$. We will use $\sim$ for negation, and the following binary connectives, given in order of decreasing binding strength: & (extensional conjunction), $\circ$ (fusion, or intensional conjunction), $\vee$ (extensional disjunction), + (fission, or intensional disjunction), and $\rightarrow$ (implication). We will use $A, B, C, D, E, A_1, \ldots$ as schematic variables ranging over formulas. Our bracketing conventions are those of [10] augmented where necessary by those of [1], and we will occasionally suppress brackets where the precedence of connectives makes the meaning clear. Formulas which are either propositional variables or single negations of them will be called *literals*. We also define a connective for logical equivalence: $A \Leftrightarrow B =_{df} (A \rightarrow B) \& (B \rightarrow A)$. We now provide a Hilbert-style axiomatization of **LR**:

A1.	$A \rightarrow A$	A9.	$(A + B) \Leftrightarrow (\sim A \rightarrow B)$
A2.	$A \rightarrow B \rightarrow . C \rightarrow A \rightarrow . C \rightarrow B$	A10.	$A \& B \rightarrow A$
A3.	$A \rightarrow B \rightarrow . B \rightarrow C \rightarrow . A \rightarrow C$	A11.	$A \& B \rightarrow B$
A4.	$(A \rightarrow . A \rightarrow B) \rightarrow . A \rightarrow B$	A12.	$(A \rightarrow B) \& (A \rightarrow C) \rightarrow . A \rightarrow B \& C$
A5.	$(A \rightarrow . B \rightarrow C) \rightarrow . B \rightarrow . A \rightarrow C$	A13.	$A \rightarrow A \vee B$
A6.	$\sim \sim A \rightarrow A$	A14.	$B \rightarrow A \vee B$
A7.	$A \rightarrow \sim A \rightarrow \sim A$	A15.	$(A \rightarrow C) \& (B \rightarrow C) \rightarrow . A \vee B \rightarrow C$
A8.	$A \rightarrow \sim B \rightarrow . B \rightarrow \sim A$	A16.	$A \rightarrow . B \rightarrow . A \circ B$
		A17.	$(A \rightarrow B \rightarrow C) \rightarrow . (A \circ B) \rightarrow C$
R1.	If $\vdash A$ and $\vdash A \rightarrow B$ then $\vdash B$	R2.	If $\vdash A$ and $\vdash B$ then $\vdash A \& B$

(2) For those familiar with relevant logics, the only thing new about **LR** is the name. It is perhaps most familiar as **R**-dist, and is just the principal relevant logic **R** minus the distribution axiom. The name is a contraction of the term "lattice-relevance".

Note that some of these axioms are redundant. From this axiomatization of **LR** we obtain the principal relevant logic **R** by simply adding the following axiom:

A18. $A \& (B \vee C) \rightarrow (A \& B) \vee (A \& C)$ Distribution

From this axiomatization of **R** we can obtain the classical propositional logic **TV** by adding the (paradoxical) implicational axiom

A19. $A \rightarrow . B \rightarrow A$

Finally, we note without proof that the following are all theorems of **LR**

T1.	$\sim(A \vee B) \Leftrightarrow (\sim A \& \sim B)$	T4.	$\sim(A \circ B) \Leftrightarrow (\sim A + \sim B)$
T2.	$\sim(A \& B) \Leftrightarrow (\sim A \vee \sim B)$	T5.	$\sim(A + B) \Leftrightarrow (\sim A \circ \sim B)$
T3.	$(A \rightarrow B) \Leftrightarrow (\sim A + B)$	T6.	$\sim(A \rightarrow B) \Leftrightarrow (A \circ \sim B)$

and that the connectives &, $\vee$, $\circ$ and + are each provably associative and commutative in **LR**. The connectives & and $\vee$ differ from their intensional counterparts $\circ$ and + respectively in that whereas & and $\vee$ are idempotent in **LR**, $\circ$ and + are not. These facts may be extracted from [1] (p. 396).

Despite some of the logical oddities of **LR**, we believe that the fact that **LR** and **R** differ only over the so-called extensional axioms and so share all the relevant insights, and the fact that **LR** is the largest natural fragment of **R** known to be decidable, makes it a good starting place for any investigation of the possibilities for relevant ATP. KRIPKE, apart from being an automated theorem-prover for **LR** and its proper subsystems and fragments, has also been adapted to handle other logics including the implication/negation fragment of the logic **E** described in [1], and the modalized relevant logic $\mathbf{NR_i}$ described in [19]. Most of what we have to say about ATP for **LR** applies to these other logics, and **LR** is the most intricate of the range of logics that KRIPKE can deal with. For these reasons we will confine our attention in what follows to ATP for **LR**.

Classical ATP is often based on proof methods which depend on employing clausal forms, such as Horn clause format, with the admissibility of these normal-forms relying in turn on the provability within the logic of certain equivalences. As [21] (p. 278) notes, some of the equivalences that permit appropriate normal-forms in classical logic are *not* relevantly valid; for example, relevant implication fails to distribute over disjunction and so

$$p \to q \vee r \Leftrightarrow (p \to q) \vee (p \to r)$$

is not a theorem of relevant logic. Hence KRIPKE cannot proceed on the basis of these standard clausal forms. However, inspection of T1-T6 and A6 suggests that we can achieve *negation-normal-form* on **LR** formulas, eliminating $\to$ in favour of $\circ$ and $+$, and restricting $\sim$ to propositional variables; this is indeed the case. For ease of presentation, we will assume in what is to follow that all **LR** formulas are in negation-normal-form.

Negation-normal-form will not help us with our problems in using clausal-based techniques. Non-clausal techniques and approaches to ATP for classical logics, or fragments thereof, are not unknown however, and in some circles these techniques are preferred. There has been some research into ATP using Beth-Smullyan analytic methods, for which see [24] and [31], and quite a comprehensive survey of other, mostly earlier, non-resolution theorem proving given by Bledsoe in [9]. Bledsoe expressed a preference for "natural" or "goal-directed" techniques, attributable to Gentzen, and predicted that these techniques would grow in popularity. This prediction would seem, in the light of [26], [7], [5] and [6], to be fairly accurate. Gentzen-style proof-theoretic systems have been, as it happens, very important in the development of relevant logics since the late 1950,s and our development herein of ATP methods for **LR** will be based on Gentzen-style consecution formulations of the logic. Our preferred terminology for detailing this proof-theoretical treatment of **LR** will be the *production system* terminology of [27].

Our initial proof-theory for **LR**, which we call **L1**, is due to [14], [4] and [19]. Our *global database* will be a *multiset* of formulas (i.e. sequences that permit free permutation of members; alternatively, as sets that allow repetitions – see [22] for a discussion). Given some arbitrary (finite) collection of formulas we shall formally represent the multiset containing them by enclosing them in double square brackets. Thus $[\![A, A, A, B, C, C]\!]$ is a multiset containing three occurrences of A, one of B and two of C. We will let

$$\alpha, \beta, \gamma, \delta, \varepsilon, \alpha_1, \ldots$$

be variables ranging over multisets. Our axioms or *goals* will be multisets of the form $[\![A, \sim A]\!]$, which in light of the deduction theorem presented in [1], can be interpreted as formulas of the form $(A + \sim A)$. For ease of presentation, we will drop the multiset braces $[\![$and$]\!]$, where context makes the meaning clear.

Before detailing the *production rules* that define **L1**, we introduce some terminology for dealing with multisets. We let c be a function over multisets such that $c(A;\alpha)$ gives the count of the number of times the formula A occurs in multiset α; thus, $c(B;[\![A,B,C,B,D]\!])=2$. Every multiset is associated in a natural way with a certain set – namely the set containing just the members of the multiset. We shall call the members of this set the *generators* of the multiset, and the multiset consisting just of the generators of α the *base-multiset* of α. We will say that a multiset α is a *sub-multiset* of β iff for every formula A in β, $0 \leq c(A;\alpha) \leq c(A;\beta)$. Moreover, if α is a sub-multiset of β, and the base-multisets of α and β are identical, then ve say that β *subsumes* α.

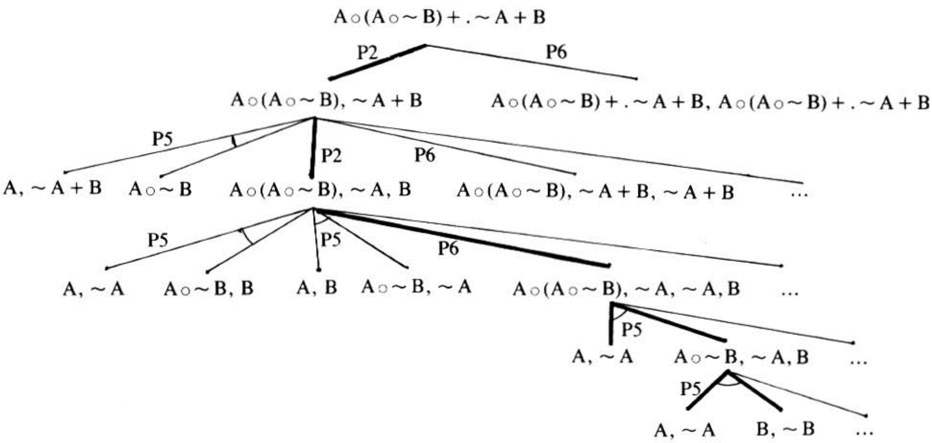
We now provide the production rules that define **L1**:

P1.	$[\![A \& B,\alpha]\!]$	$\mapsto$	$[\![A,\alpha]\!]$; $[\![B,\alpha]\!]$	
P2.	$[\![A + B,\alpha]\!]$	$\mapsto$	$[\![A,B,\alpha]\!]$	
P3.	$[\![A \vee B,\alpha]\!]$	$\mapsto$	$[\![A,\alpha]\!]$	
P4.	$[\![A \vee B,\alpha]\!]$	$\mapsto$	$[\![B,\alpha]\!]$	
P5.	$[\![A \circ B,\alpha]\!]$	$\mapsto$	$[\![A,\beta]\!]$; $[\![B,\gamma]\!]$	where for all C in α, $c(C;\alpha) = c(C;\beta) + c(C;\gamma)$
P6.	$[\![A,\alpha]\!]$	$\mapsto$	$[\![A,A,\alpha]\!]$	

Each rule is to be interpreted as saying that the left-side multiset is provable if the right-side multiset(s) is (are both) provable, and so P1-P6 provide a way of *decomposing* an initial multiset into subgoals. By applying all possible rules to an initial multiset, and then in turn to the subgoals so generated, one generates an *AND/OR tree* of multisets in the manner of [27]. One may then search this tree to see if it contains a *solution tree* or *proof* for the initial multiset; i.e. a tree of complete connectors, the tips of which are axioms or goal nodes of the form $[\![A, \sim A]\!]$. The following illustrates a segment of the search tree for the formula

$$A \circ (A \circ \sim B) + . \sim A + B$$

which is the negation-normal-form of axiom A4. AND-nodes have their connectors arced, connectors are annotated with the production rule that produced them, and the solution tree has its connectors bolded.



Initial Segment of an AND/OR Search Tree

P1-P6 are all classically sound rules. Rule P6 permits free duplication of any member of a multiset into a subgoal, and corresponds to the *contraction* principle, A4. Note especially that relevant logics do *not* admit the following rule

P7. $[\![A, \alpha]\!] \mapsto [\![\alpha]\!]$

which corresponds to the *weakening* principle of [12]. P7 is however classically sound, and would typically be used to reduce a multiset of the form $[\![A, \sim A, \alpha]\!]$ to a goal multiset of the form $[\![A, \sim A]\!]$, eliminating what is essentially *irrelevant* information in α which could not otherwise be *used* by the other rules to effect a proof of the initial multiset. (3)

(3) The admissibility of P7 in classical logic is part of the reason that classical logic is *monotonic*. If α is a previously provable initial multiset, then adding more formulas to α will still result in a provable multiset because P7 can be used to ignore these additions. **LR**, and **R** for that matter, are *non-monotonic* logics. See [37] for details.

On the other hand, the following rule which is a restricted version of P7

P8. $[\![\alpha, A, A, B, \sim B]\!] \mapsto [\![\alpha, A, B, \sim B]\!]$

is admissible in **L1**, on the basis of an observation by Meyer that

$$(C \rightarrow . A \rightarrow . B \rightarrow B) \Leftrightarrow . C \rightarrow . A \rightarrow . A \rightarrow . B \rightarrow B$$

is a theorem of **LR**. P8 licences the *simplification* of multisets that contain a goal multiset as a sub-multiset. Repeated applications of P8 *reduce* a multiset of the form $[\![\alpha, B, \sim B]\!]$ to one of the form $[\![\beta, B, \sim B]\!]$, where β is the base-multiset of α. Where α itself contains a goal multiset as a sub-multiset, further reductions may also be possible; e.g., $[\![A, A, \sim A, B, B, \sim B]\!]$ first reduces to $[\![A, \sim A, B, B, \sim B]\!]$, and then to $[\![A, \sim A, B, \sim B]\!]$. If a multiset α is reduced to β, the copies of the generators of α that have been deleted in the move to β can be subsequently replaced, if so desired, by using the P6 rule.

Note that the rule

P9. $[\![\alpha, \beta]\!] \mapsto [\![\alpha, A]\!]; [\![\beta, \sim A]\!]$

is also admissible in **L1**. P9 is a variant of Gentzen's *Haupsatz*, and the proof that P9 is admissible in **L1** is the basis of the proof by [14], [4] and [19] that **L1** is sound and complete w.r.t. **LR**; i.e., formula A is a theorem of **LR** iff there is an **L1** solution tree for $[\![A]\!]$. The problem with **L1** is that in the presence of the rule P6, the search tree for any formula will be infinite. Thus, any algorithm that searches this tree for a solution tree may not terminate. We can however, following Kripke in [14] and Meyer in [19], modify the rules of **L1** and impose conditions on search trees so that all search trees will be finite, and termination thus guaranteed. We shall call this modified system **L2**.

L2 eliminates P6 by building its effect into the other five rules. **L2** is formulated with P1-P4, and adds the following rules:

P1′. $[\![A \& B, \alpha]\!] \mapsto [\![A, A \& B, \alpha]\!]; [\![B, A \& B, \alpha]\!]$
P2′. $[\![A + B, \alpha]\!] \mapsto [\![A, B, A + B, \alpha]\!]$
P3′. $[\![A \vee B, \alpha]\!] \mapsto [\![A, A \vee B, \alpha]\!]$
P4′. $[\![A \vee B, \alpha]\!] \mapsto [\![B, A \vee B, \alpha]\!]$

P5′. $[\![A \circ B, \alpha]\!] \mapsto [\![A, \beta]\!]; [\![B, \gamma]\!]$ where (i) β and γ are both sub-multisets of $[\![A \circ B, \alpha]\!]$ (ii) for all C in α, $c(C;\alpha) \leq c(C;\beta) + c(C;\gamma)$.

P5 is clearly a special case of P5′. But P5′ includes the extra possibilities of taking a copy of the decomposed formula ($A \circ B$) into the first subgoal, the second subgoal, or both, and the possibility of placing each member of α not merely into one subgoal or the other, but possibly both. Note that P8 is still admissible in **L2**.

Kripke and Meyer showed that **L2** was sound and complete w.r.t. **LR**, and moreover, that by imposing the following condition on **L2** search trees (due essentially to [11]), all **L2** search trees are provably finite. The condition is a variant of the *acyclicity constraint* that Nilsson places on AND/OR search trees, wherein identical databases are not permitted on the same branch of any tree. The constraint on **L2** search trees, which we call the *Curry condition*, is that for all multisets α in a search tree, no multiset β in the upward path from α should be subsumed by α. (4)

The finitude of **L2** search trees and the fact that **L2** is sound and complete w.r.t. **LR** entails that **LR** is decidable. But only just! Kripke, in a private communication to McRobbie, conjectures that his decision procedure is not a primitive recursive function. Although there is an upper bound on the number of *immediate* subgoals to any given (finite) multiset α, namely

$(4k \cdot 3^{n-1}) + 4l + 2m$

where n is the number of formulas in α, k the number of fusion formulas, l the number of disjunctive formulas, and m the number of conjunctive and fission formulas in α, (5)

there would seem to be no nice way of predicting when a given

(4) This constraint can not be placed on **L1** search trees; to do so would prevent any application of the P6 rule to a multiset α, as the subgoal produced by applying P6 to α always subsumes α.

(5) Note that this is a rough upper bound, and that if duplicate subgoals are collapsed, the number of distinct immediate subgoals will often be somewhat smaller.

rule-application will generate a subgoal that subsumes a multiset in its upward path. Moreover, consider the multiset $[\![A \circ B, \alpha]\!]$. One of its possible pairs of subgoals is $[\![A, A \circ B, \alpha]\!]$ and $[\![B, A \circ B, \alpha]\!]$. In this case the subgoals are, in a sense, more complex than the initial multiset. And in general, the size and complexity of multisets in a branch of the search tree may grow horrendously until the requirements of the Curry condition eventually terminate the growth of the branch. It takes little imagination to see that, even with quite simple α, the number of multisets in the **L2** search tree for α can be staggeringly high.

3. *Efficient ATP Techniques for* LR

The version of KRIPKE that we discuss in [16] used the **L2** rules, and it included various demons (which at the time we called *filters*) which checked multisets as they were introduced into a search tree to ensure that they meet certain necessary (though not sufficient) conditions for provability. A multiset that did not meet these conditions could not be part of a solution tree for the initial multiset, and so was not expanded.(6) We shall describe these demons later on. Suffice to say that this version of KRIPKE, while being far from impotent, proved incapable of solving some of the more difficult problems that concerned us and which we discuss in Section 4. In true dialetic fashion, the output from this version of KRIPKE suggested that **LR** could be captured by a more efficient set of rules, which prompted Thistlewaite in [37] to propose the following set of rules.

Let **L3** be the set of rules including P1-P4, and

P3″. $[\![A \vee B, \alpha]\!] \mapsto [\![A, A \vee B, \alpha]\!]$ where $A \vee B$ does not occur in α.

P4″. $[\![A \vee B, \alpha]\!] \mapsto [\![B, A \vee B, \alpha]\!]$ where $A \vee B$ does not occur in α.

(6) This is somewhat historically inaccurate, in that some of these conditions were only discovered in the work of [37], and so were only included in versions of KRIPKE based on this work.

P5″. $[\![A \circ B, \alpha]\!] \mapsto [\![A, \beta]\!] ; [\![B, \gamma]\!]$ where
(i) if $A \circ B$ not in α, then it *may* occur in β, γ or both
(ii) for all C in α, if $c\,(C;\alpha > 1$ then $c\,(C;\alpha) = c\,(C;\beta) + c\,(C;\gamma)$
else C *must* occur in β, γ or both.

Note that P1′ and P2′ have been eliminated, and P3″-P5″ result from placing severe restrictions on P3′-P5′. Whereas P3′-P5′ built the full effect of P6 into these rules, allowing free copying of a decomposed compound formula into a subgoal, P3″-P5″ only allow this where the subgoal would not otherwise have *any* copy of the decomposed formula. Unlike P5′, P5″ closely resembles P5 itself: one must at least partition α into β and γ, but optionally can take one copy of a generator of α into β or γ provide that generator does not already occur there. The motto is that a single copy of any generator is (logically) sufficient in a multiset.

Thistlewaite in [37] proved that **L3** is indeed sound and complete w.r.t. **LR**, and that if the Curry condition is imposed on **L3** search trees, then these trees will be finite. Soundness follows simply from the obseration that, as the **L3** rules are sub-cases of the **L2** rules every **L3** solution tree will be an **L2** solution tree, and **L2** is sound w.r.t. **LR**. The proof of completeness depends on certain *structural* facts about search trees constructed by the **L3** rules, but as the Curry condition is itself a structural property of search trees, the proofs of completeness and decidability interact in complex ways. These proofs can be found in [37].

Two of these structural facts are of independent computational interest: if a multiset α of the form $[\![A \,\&\, B, \beta]\!]$ has an **L3** solution tree then there is a solution tree for α in which the immediate subgoals for α are generated by an application of P1; and if α is of the form $[\![A + B, \beta]\!]$ and is soluble, then there is a solution tree for α in which the immediate subgoal for α in this tree comes via P2. In the logical parlance of [12], P1 and P2 would be said to be *invertible* rules; in the computational parlance of [27], they would be said to be *irrevocable* rules. In [37], it is shown that these facts can be built into the **L3** rules by adding the following condition to each of P3, P4, P3″, P4″ and P5″: α must not contain either a conjunction or a fission formula as an explicit member.

To illustrate the computational superiority of the **L3** rules over the **L2** rules, we note that many multisets under the **L3** rules will have only one or two immediate subgoals, because of the invertability of P1 and P2. Multisets of the form ⟦A ∘ B, α⟧ will typically generate the greatest number of possible immediate subgoals, but even here, the differences between **L3** and **L2** are dramatic. For example, ⟦p ∘ q, ∼ p, ∼ p, ∼ q⟧ has 72 distinct subgoals in **L2**, but only half as many in **L3**. As the number of repetitions of generators in a multiset increases, the relative savings increase spectacularly; for example, q, ∼ q, ∼ q⟧ has 800 distinct subgoals in **L2** but only 48 in **L3**. Even so, the branching factor of the **L3** rules is high enough for an unconstrained application of them to often cause trouble. Quite often, too, a multiset in a search tree will have only one solution path, and so it is imperative that specialized knowledge about relevantly provable multisets be used to avoid expanding insoluble multisets.

We mentioned at the start of this section that there are various necessary conditions that any multiset must meet if it is to be provable. The first of these conditions is known in the logical literature either as the 'positive-negative parts property' or the 'antecedent-consequent parts property' (see [1]), and it amounts to a relevant analogue of the Davis-Putnam *pure literal rule*. The positive parts of a multiset are just the unnegated propositional variables that occur as part of any formula in the multiset, and the negative parts are just the propositional variables that occur negated, where all members of the multiset are understood as being in negation-normal-form. Thus, the set of positive parts of ⟦A & (∼ B ∘ C), ∼ B, C + ∼ A⟧ is {A, C} and the set of negative parts is {A, B}. A multiset α is said to have the *strong positive-negative parts property* iff the set of positive parts of α is identical with the set of negative parts of α. A multiset α is said to have the *weak positive-negative parts property* iff at least one propositional variable occurs as both a positive and negative part of α. We now have the following condition due essentially to Maksimova, Anderson and Belnap:

> *Parts Condition*: If a multiset α contains no disjunction as a part of any formula in α, then α has a solution (i.e. is provable) only if α has the strong positive-negative parts property. Otherwise, α has a solution only if α has the weak positive-negative parts property. The proof can be extracted from [1] (pp. 253-254).

Unlike the pure literal rule of Davis and Putnam, a multiset's failing the parts condition does not license the *reduction* of the multiset. We are not permitted to delete multiset members that have no matching positive or negative part (as with the pure literal rule), for this involves tacit use of the generalized weakening principle, P7. While on the subject of the Davis-Putnam rules for classical ATP, we note that *all* are inadmissible strategies within relevant ATP, as they all involve tacit use of P6. The pure literal rule does have an analogue – the parts condition – but the others appear to have no relevant analogues.

We have noted that the effect of P6, in the absence of P7, makes for many combinatorial problems in applying the P5″ rule, and we have just noted that the absence of P7 rules out the Davis-Putnam deletion strategies within relevant ATP. We now note, though, a property of relevantly provable multisets related to the parts condition, which demonstrates a *computational advantage* due to the absence of P7. The advantage of not having P7 centers on the sensitive goal-directedness that one would expect from relevance requirements such as variable-sharing.

> *Strict Parts Condition*: A multiset α has a solution only if the set of positive (negative) parts of the *compound formulas* of α contains the set of negated (unnegated) propositional variables which are *explicit members* of α, or α is an axiom. Again, see [37] for a proof of this claim.

For example, the multiset $\alpha = [\![p \circ \sim q, r \& s,, q, \sim r]\!]$ meets the strict parts condition, whereas $\beta = [\![p \circ \sim q, r \& s, \sim r, \sim q]\!]$ does not meet the condition, because the explicit negated variable, $\sim q$, has no unnegated mate, q, as a positive part of some compound formula in β. In the context of constructing search trees, the goal-directing nature of the strict parts condition on multisets becomes clear. Once a literal becomes an *explicit* member of some multiset α in some branch of the search tree due, say, to the decomposition of some compound formula into simpler parts, then because P7 is inadmissible, the literal can not be ignored – if α is to have a solution, the literal will eventually have to end up in some goal multiset consisting of it together with its complementary literal. But all of the rules of **L3** decompose compound formulas, and so any multiset β in the subtree on α will have a part of some compound member of α in β. Note that a multiset may meet the strong or weak positive-negative parts conditions, and yet fail to meet the strict parts condition; e.g., $[\![p \circ \sim p, q, \sim q]\!]$. Looking at the rules of **L3**, it is interesting to observe that the *only* way a part of some formula in an initial multiset can 'disappear' part way down the branch of some search tree is for it to be a disjunct, the parent disjunction of which was decomposed using P3 or P4 (indeed, this is the reason for the split between the strong and weak positive-negative parts conditions). Prohibiting the expansion of multisets that fail to satisfy the strict parts condition has proven to be an extremely effective node-pruning device in KRIPKE.

On certain multisets – namely, those that contain no fusion formula as any part of any member of the multiset – we can impose the following condition for them to be eligible for expansion:

> *Rule-of-2 Condition*: A fusion-free multiset α is provable only if α has no more than 2 explicit members. A proof of this can be found in [37].

Another condition relies on a semantical interpretation of multisets within algebraic structures that model **LR**. We assume the reader has some knowledge of algebraic structures for propositional logics, otherwise known as *logical matrices* or *n-valued truth-tables*. A matrix for **LR** is a structure of the form $M = \langle \mathcal{M}, \mathcal{O}, \mathcal{D} \rangle$, where $\mathcal{M}$ is some set of elements, $\mathcal{D}$ some subset of $\mathcal{M}$ consisting of the so-called *designated elements* of $\mathcal{M}$, and $\mathcal{O}$ some set of n-ary operations on $\mathcal{M}$, with the operations of $\mathcal{O}$ corresponding 1-1 to the connectives in the propositional language underlying **LR**. A *model, m,* of some formula A in some matrix M is just a mapping from the propositional variables in A to members of $\mathcal{M}$, extended to a homomorphism into M by interpreting the connectives in A via the corresponding operations in $\mathcal{O}$. Formula A *holds* in m iff $m(\mathrm{A}) \in \mathcal{D}$, and is *valid* in M iff it holds in all models (i.e., under all assignments of members of $\mathcal{M}$ to the variables in A). An **LR**-matrix *satisfies* **LR**, in the sense that all **LR** axioms A1-A17 are valid in the matrix and $\mathcal{D}$ is closed under the rules R1 and R2. (7)

(7) Unlike **TV**, **LR** has no finite characteristic matrix – i.e., a finite matrix which validates all and *only* the theorems of the logic.

We can now state the condition:

> *Matrix Condition*: Let $\mathscr{S} = \{M_1,\ldots,M_n\}$ be some set of logical matrices satisfying **LR**, α be any multiset, and α^+ be the formula given by compounding the members of α using the fission connective, $+$. Then α has a solution only if α^+ is valid in all members of $\mathscr{S}$. Further details can be found in [37].

While testing multisets for the parts conditions and the rule-of-2 condition is a computationally fast procedure, testing for the matrix condition is an exponential problem. It is well-known that where a matrix M has k elements in $\mathscr{M}$, and α^+ has n distinct propositional variables, that there are k^n models of α^+ in M. However, within any set $\mathscr{S}$ of matrices there will often be a large number of redundant models of an n-variable formula, because of various homomorphisms between and within matrices in $\mathscr{S}$ and because of the peculiar properties of the n-ary function spaces of these matrices. This informal notion of a redundant model is thoroughly discussed in [37]; suffice to say that within the class of **LR**-matrices having 10 or fewer elements, only 5% of the total number of 1-variable models are non-redundant, and only 40% of 5-variable models are non-redundant. Similar savings occur across the board. As a consequence, $\mathscr{S}$ can in practice contain a sizeable number of relatively large matrices while being computationally manageable. [37] also discusses other techniques that are employed for reducing the overheads of checking for the matrix condition, including various criteria for selecting the membership of $\mathscr{S}$, ways of normalizing α^+ relative to $\mathscr{S}$ and α and the location of α within the search tree, and a few tricks at the operational level for testing formulas for validity in a matrix.

In the process of creating and searching a search tree for some initial multiset α, we will of course discover that certain of the subgoals in the tree have solution trees, and that other subgoals do not, and we can keep respective lists of such multisets. We can use this information dynamically, and as a multiset is expanded we can check the multisets introduced into the search tree to see if their provability or unprovability is already known. We also know that if a multiset α is provable, then so too is every multiset that is subsumed by α; indeed, this is the logical support for imposing the Curry condition. Conversely, if α is unprovable, then so too is every multiset that subsumes α. As a consequence, our list of known-provable multisets need only contain multisets which are *maximal* on a set of generators, where a multiset is so maximal iff no other multiset in the list subsumes it; and conversely, our list of known-unprovable multisets need only contain multisets which are *minimal* (in the obvious sense) on a set of generators.

4. *Problems and Proofs*

KRIPKE's worth lies not simply in its breaking ground, nor does it lie simply in being a handy desk-top logicians' helper (although neither are unimportant uses, as [41] rightly notes). But to the point, we mention briefly two particular problems – one from within ATP, and the other from within logic – that KRIPKE has been brought to bear fruitfully upon.

In [28], Ohlback and Wrightson provide a solution to what is termed "Belnap's Problem", concerning the provability of the following formula in **R**. Further details of the problem are given in [29].

$$\text{WF} \qquad A \rightarrow (B \rightarrow B)) \rightarrow (A \rightarrow (A \rightarrow (B \rightarrow B)))$$

Note that WF is also a theorem of **LR**. The problem is somewhat misrepresented in that the provability of WF is of no special interest in relevant logic, nor is it difficult (even for a human) to prove. Its real interest lies, we believe, in the fact that it gives a way of comparing relevant ATP based on proof techniques not specialized to relevant logics, with ATP using the specialized proof procedures of KRIPKE. [28] proved WF using (a modified version of) the Markgraf Karl Refutation Procedure; for details of which, see [8]. The Markgraf Karl Refutation Procedure is a first-order classical ATP system, and the proof that WF was **R**-provable was effected by treating the propositional relevant logic **R** as a first-order theory, using the Meyer-Routley semantics given in [34] for **R** to define the theory. Given our earlier discussion of higher-order classical approaches to non-classical theorem proving, it was not surprising that the approach of [28] presented a proof of WF in around 10 minutes of CPU time, while KRIPKE can prove WF in about 1/10th of a second. KRIPKE does not access any of its special global knowledge, such as that detailed in the previous section about relevant provability, to effect this proof of WF; it follows fairly immediately on an unconstrained application of the appropriate rules. Wrightson also informs us that, to date, the proof in [28] that WF is a theorem of **LR** and **R** has not been matched by any other classical ATP system. This result clearly supports our claim that relevant ATP systems employing relevant proof procedures will perform more efficiently than generalized systems. We describe now a problem which was of considerable logical interest to which KRIPKE was applied.

KRIPKE was the cornerstone of an attack on the decision problem for **R**, details of which can be found in [17]. This work was interrupted by Urquhart's brilliant proof in [38] that **R** was in fact *undecidable*.[39] discusses some of the ramifications of this result, including several unforeseen connections between relevant implication and projective geometry, and notes that **R** is perhaps the first independently motivated, undecidable propositional logic. The question as to whether **R** was decidable or not had been an open problem in logic for 25 years, and was of enormous logical difficulty. While Urquhart's proof utilized innovative associations of ideas, far beyond the conception of [17], the two approaches shared the basic premiss that **R** was undecidable and that this undecidability could be related to the definability of an appropriately free associative connective within **R** to act like the desired semigroup operation in the manner of [32].

Our version of this approach was to generate candidate definitions for such a connective, and prune the list of these candidates by showing that certain of them were *not* appropriately free – in particular, if a candidate definition defined a connective which was provably associative in **LR**, then the fact that **LR** is decidable would mean that the condidate definition could not be sufficiently free in **R**. The second stage of our approach involved examining the remaining candidates (if any) to see if one of them in fact defined an appropriately free associative connective in **R**, using the techniques of [23]. Some of the more plausible candidate definitions are listed below. KRIPKE featured in the first stage of our attack on the decision problem for **R**; that is, KRIPKE assisted in establishing that each of our best candidate definitions, C1-C16, does in fact define an associative connective in **LR**.

Let the binary connectives $\oplus_i$, $1 \leq i \leq 16$, be defined by the corresponding 16 formulas:

C1.	$A \circ (A + A) \circ (B \vee \sim B)$	C9.	$A \circ (A + (\sim A \circ (B \vee \sim B)))$
C2.	$A \circ (B \vee \sim B \vee (\sim A \circ \sim A))$	C10.	$A \circ (A \vee (\sim A \circ (B \vee \sim B)))$
C3.	$A + (A \circ \sim A \circ (B \vee \sim B))$	C11.	$A \vee (A \circ (\sim B \vee (\sim A \circ B)))$
C4.	$A \vee (A \circ \sim A \circ (B \vee \sim B))$	C12.	$A \circ (\sim B \vee (A + (\sim A \circ B)))$
C5.	$A \circ (A \vee \sim A) \circ (B \vee \sim B)$	C13.	$A \circ (\sim B \vee (B \circ (A \vee \sim A)))$
C6.	$A \circ (A \vee \sim B \vee (\sim A \circ B))$	C14.	$A \circ (\sim B \vee (B + (\sim A \circ B)))$
C7.	$A \circ B \vee \sim B \vee (\sim A \circ B))$	C15.	$A \circ (\sim B \vee (B \circ (\sim A \vee B)))$
C8.	$(A + A) \circ (\sim B \vee (A \circ B))$	C16.	$(A \vee (A \circ \sim A)) \circ (B \vee \sim B)$

The question of whether these define associative connectives in **R** or **LR** then amounts to whether each of $(C \oplus_i (D \oplus_i E)) \Leftrightarrow ((C \oplus_i D) \oplus_i E)$ is provable in **R** or **LR**. For example, in the case of $i = 16$, this amounts to whether the following formula is provable:

$$((((((C \vee (C \circ \sim C)) \circ (D \vee \sim D)) \vee (((C \vee (C \circ \sim C)) \circ (D \vee \sim D)) \circ \sim ((C \vee (C \circ \sim C)) \circ (D \vee \sim D)))) \circ (E \vee \sim E))$$
$$\rightarrow ((C \vee (C \circ \sim C)) \circ (((D \vee (D \circ \sim D)) \circ (E \vee \sim E)) \vee ((D \vee (\circ \sim D)) \circ (E \vee \sim E)))))\ \&$$
$$(((C \vee (C \circ \sim C)) \circ (((D \vee (D \circ \sim D)) \circ (E \vee \sim E)) \vee ((D \vee (D \circ \sim D)) \circ (E \vee \sim E)))) \rightarrow$$
$$((((C \vee (C \circ \sim C)) \circ (D \vee \sim D)) \vee (((C \vee (C \circ \sim C)) \circ (D \vee \sim D)) \circ \sim ((C \vee (C \circ \sim C)) \circ (D \vee \sim D)))) \circ (E \vee \sim E))))$$

Although the question of the decidability of **R** has been solved, and our attack on it has thus since lapsed, we note that KRIPKE can now prove at least one direction of associativity for the connectives defined by most of the C1-C16, and has complete proofs of associativity for some of them. The runtimes vary, and range up to about 20 minutes CPU time. The best time is for the complete proof of the associativity of the connective defined by C16, which KRIPKE provides in about 90 seconds. The proofs are, as the reader would expect, quite long. We commend C1-C16 and the task of proving them associative in **R** or **LR**, to the ATP community as a means of measuring the problem-solving strength of various ATP systems. On the basis of the respective performances of KRIPKE and the system of [28] at proving WF, we conjecture that extent general purpose first-order ATP systems will have considerable difficulty in proving the associativity of *any* of the connectives defined by C1-C16.

Australian National University
Dept. of Philosophy
GPO Box 4, Canberra ACT 2601
Australia

P.B. THISTLEWAITE,
R.K. MEYER and M.A. MCROBBIE

REFERENCES

[1[Anderson A.R. and Belnap N.D., Jr. *Entailment: The Logic of Relevance and Necessity, Vol. 1.* Princeton University Press, New Jersey, 1975.

[2] Belnap N.D., Jr. How a Computer Should Think. In *Contemporary Aspects of Philosophy*, Ryle, G., Ed., Oriel Press, 1977, pp. 30-55.

[3] Belnap N.D., Jr. A Useful Four-Valued Logic. In *Modern Uses of Multiple-Valued Logic*, Dunn J.M. and Epstein G., Ed., Reidel, 1978, pp. 8-37.

[4]Belnap N.D. Jr. and Wallace J.R., "A Decision Procedure for the System E_I of Entailment with Negation". *Zeitschrift fur mathematische Logik und Grundlagen der Mathematik 11* (1965), 277-289.

[5] Bibel W., "A Theoretical Basis for Systematic Proof Procedure". *Lecture Notes in Computer Science 88* (1980), 154-167.

[6] Bibel W., *Automated Theorem Proving.* Vieweg, Braunschweig, 1982.

[7] Bibel W. and Schreiber J., Proof Search in A Gentzen-Like System of First Order Logic. In *Proceedings of the International Computing Symposium*, North Holland, Amsterdam, 1975, pp. 205-212.

[8] Blasius K., Eisenger N., Siekmann J., Smolka G., Herold A. and Walther C., The Markgraf Karl Refutation Procedure (Fall 1981). Interner Bericht, University of Karlsruhe.

[9] Bledsoe W.W., "Non-Resolution Theorem Proving". *Artificial Intelligence 9* (1977), 1-35.

[10] Church A., *Introduction to Mathematical Logic.* Princeton University Press, New Jersey, 1956. 6th printing, 1970.

[11] Curry H.B., *A Theory of Formal Deducibility.* Ann Arbor, Michigan, 1950. 2nd edition 1955, reprinted 1957.

[12] Curry H.B., *Foundations of Mathematical Logic.* McGraw-Hill, New York, 1963. Corrected reprint, Dover, New York, 1977.

[13] Haspel C.H., *A Study of Some Interpretations of Modal and Intuitionistic Logics in the First Order Predicate Calculus.* Ph. D. Th., Syracuse University, New York State, 1972.

[14] Kripke S.A., "The Problem of Entailment". *Journal of Symbolic Logic 24* (1959), 324. Abstract.

[15] Lewis C.I., *A Survey of Symbolic Logic.* University of California, Berkeley, 1918. Reprinted 1961, Dover.

[16] McRobbie M.A., Meyer R.K. and Thistlewaite P.B., Computer-Aided Investigations into the Decision Problem for Relevant Logics: A Preliminary Report. Unpublished. Presented to the *5th Conference on Automated Deduction* held at Les Arcs, France.

[17] McRobbie M.A., Meyer R.K. and Thistlewaite P.B., Computer-Aided Investigations into the Decision Problem for Relevant Logics: The Search for a Free Associative Connective. In *Australian Computer Science Communications*, Goldschlager L.M., Ed., Basser Dept. of Computer Science, University of Sydney, 1983, pp. 236-267. Proceedings of the 6th Australian Computer Science Conference.

[18] McRobbie M.A., Thistlewaite P.B. and Meyer R.K., "A Mechanized Decision Procedure for Non-Classical Logics: The Program KRIPKE". *Journal of Symbolic Logic 47* (1982), 717. Abstract.

[19] Meyer R.K., *Topics in Modal and Many-Valued Logic*. Ph. D. Th., University of Pittsburgh, Pennsylvania, 1966.

[20] Meyer R.K., A Boolean-valued Semantics for R. Research Paper 4, Logic Group, Research School of Social Sciences, Australian National University, Canberra, 1978.

[21] Meyer R.K., "Almost Skolem Forms for Relevant (and other) Logics". *Logique et Analyse 24* (1981), 277-289.

[22] Meyer R.K. and McRobbie M.A., "Multisets and Relevant Implication I and II". *Australasian Journal of Philosophy 60* (1982), 107-139 and 265-281.

[23] Meyer R.K. and Routley R., "An Undecidable Relevant Logic". *Zeitschrift fur mathematische Logik und Grundlagen der Mathematik 19* (1973), 389-397.

[24] Mogilevski G.L. and Ostrouhov D.A., "On a Mechanical Propositional Calculus Using Smullyan's Analytic Tableau". *Cybernetics 14* (1978), 526-529.

[25] Morgan C.G., "Methods for Automated Theorem Proving in Nonclassical Logics". *IEEE Transactions on Computers C-25* (1976), 852-862.

[26] Murray N.V., "Completely Non-Clausal Theorem Proving". *Artificial Intelligence 18* (1982), 67-85.

[27] Nilsson N.J., *Principles of Artificial Intelligence*. Springer, Berlin, 1980.

[28] Ohlbach H. and Wrightson G., "Belnap's Problem in Relevance Logic". *Association of Automated Reasoning Newsletter 2* (1983), 3-5.

[29] Ohlbach H. and Wrightson G., Solving a Problem in Relevance Logic with an Automated Theorem Prover. 11, Institut fur Informatik I, University of Karlsruhe, 1983.

[30] Plaisted D.A., An Efficient Relevance Criterion for Mechanical Theorem Proving. In *Proceedings of the First Annual National Conference on Artificial Intelligence,*
American Association for Artificial Intelligence, 1980, pp. 79-83.

[31] Popplestone R.J., "Beth-Tree Methods in Automated Theorem Proving". *Machine Intelligence 1* (1967), 31-46.

[32] Post E.L., "Recursive Unsolvability of a Problem of Thue". *Journal of Symbolic Logic 12* (1947), 1-11.

[33] Quine W.V.O., *Mathematical Logic*. Harvard University Press, Cambridge, Mass., 1951. Revised Edition.

[34]Routley R. and Meyer R.K., The Semantics of Entailment I. In *Truth, Syntax and Modality*, Leblanc H., Ed., North Holland, Amsterdam, 1973.

[35] Routley R., Meyer R.K., Plumwood V. and Brady R., *Relevant Logics and Their Rivals. Part 1: The Basic Philosophical and Semantical Theory*. Ridgeview, Atascadero, California, 1984.

[36] Shapiro S.C. and Wand M., The Relevance of Relevance. 46, Computer Science Dept., Indiana University, Bloomington, Indiana, 1976.

[37] Thistlewaite P.B., *Automated Theorem-Proving in Non-Classical Logics*. Ph.D. Th., Australian National University, Canberra, 1984.

[38] Urquhart A., The Undecidability of Entailment and Relevant Implication. Forthcoming, *Journal of Symbolic Logic*.

[39] Urquhart A., Relevant Implication and Projective Geometry. Forthcoming, *Logique et Analyse*.

[40] Weyhrauch R.W., "Prolegomena to a Theory of Mechanized Formal Reasoning". *Artificial Intelligence 13* (1980), 133-170.

[41] Wos L., "Solving Open Questions with an Automated Theorem-Proving Program". *Lecture Notes in Computer Science 138* (1982), 1-21. Proceedings of the 6th Conference on Automated Deduction.