Relevance via Decomposition: A Project, Some Results, An Open Question

David Makinson
London School of Economics
david.makinson@gmail.com

Abstract

We report on progress and an unsolved problem in our attempt to obtain a clear rationale for relevance logic via semantic decomposition trees. Suitable decomposition rules, constrained by a natural parity condition, generate a set of directly acceptable formulae that contains all axioms of the well-known system R, is closed under substitution and conjunction, satisfies the letter-sharing condition, but is not closed under detachment. To extend it, a natural recursion is built into the procedure for constructing decomposition trees. The resulting set of acceptable formulae has many attractive features, but it remains an open question whether it continues to satisfy the crucial letter-sharing condition.

1 Introduction

In its standard Hilbertian axiomatization, the well-known relevance logic R has many axiom schemes and two derivation rules. The derivation rules are straightforward: conjunction (aka adjunction) $\varphi, \psi \ / \ \varphi \wedge \psi$ and detachment $\varphi, \varphi \to \psi \ / \ \psi$, where $\to$ is the non-classical propositional connective intended to represent 'relevant implication'. But the axiom schemes form a rather motley and untidy crew of about a dozen – the exact number depending on how we count, for example, the two forms of $\wedge$-elimination and $\vee$-introduction. R has also been characterized, along with several of its neighbours, by a Routley-Meyer possible-worlds semantics with three-place relations satisfying various constraints. However, notwithstanding its versatility and technical usefulness, the semantics can hardly be said to provide a satisfying rationale. Some appreciation of three-place relations may be obtained by thinking of them as indexed families of two-place relations; but there is still no intuitive rationale for the complex constraints, other than as the products of reverse engineering to ensure the validity of favoured formulae.

A more convincing perspective is provided by natural deduction which, for relevance logics, is centred on an appealing restriction of the rule of arrow introduction. However, the standard systems are still not entirely transparent, notably in their need for a 'same

suppositions' proviso on the rules of conjunction introduction and disjunction elimination. It seems rather inelegant to ensure that one connective, arrow, is well-behaved by restricting what one may do with other connectives. More significantly, the 'same suppositions' proviso leads to overkill: it has the effect of blocking derivation of the generally accepted classical principle of distribution of conjunction over disjunction, forcing its *ad hoc* addition as a separate inference rule.

It is thus natural to seek a clearer picture from a different direction, by refining the classical procedure of semantic decomposition trees. We begin the project by defining directly acceptable decomposition trees. The set of formulae validated by such trees, likewise called directly acceptable, turns out to be closed under substitution and conjunction and contains all the standard axioms of R together with certain other attractive formulae. It also satisfies the well-known letter-sharing condition, and successfully excludes other formulae which, whilst satisfying the letter-sharing condition, are notoriously repugnant for relevantists, such as 'connectivity' $(p \to q) \vee (q \to p)$ and its special case $(p \to \neg p) \vee (\neg p \to p)$, as well as $p \to (q \to p)$ and its instance 'mingle' $p \to (p \to p)$.

However, the set of directly acceptable formulae is not closed under detachment, and inspection of various examples points to a real need to ensure the closure. We do so by introducing a natural recursion into the construction of decomposition trees themselves. The resulting set of acceptable formulae has many attractive features; but it is still an open question whether it satisfies the letter-sharing condition, usually regarded as a *sine qua non* for any relevance logic worthy of the name.

It may be helpful to disclose the general methodology behind our investigation. The most common approaches to relevance logic work with Hilbertian axiom systems, Gentzen-style sequent systems, possible-worlds semantics, or natural deduction. Ours differs from all of these, although it takes some of its inspiration from natural deduction. Humberstone 2011 (page 189) articulated a widespread unease with semantic decomposition trees when he remarked that they "are not conducive to a clear-headed separation of proof-theoretic from semantic considerations". We beg to differ: one may impose syntactic constraints (in this instance, the 'crashing with parity' requirement, defined in section 4 below) on an essentially semantic procedure (decomposition trees for classical logic) and be none the less clear-headed about it. Indeed, we tend to agree with Tennant 1979 that relevance is not itself a semantic notion and there is no philosophical need to give its logic a semantics beyond that which is already provided by classical two-valued logic. Our guiding heuristic is that the logic lies in the classical semantics, while relevance is manifested in syntactic controls.

2 Recalling Classical Decomposition Trees

We recall briefly the use of semantic decomposition trees, also known as semantic tableaux, in classical propositional logic with formulae in the connectives $\wedge, \vee, \neg$ Given a formula φ

one can test whether it is a tautology by building a tree whose root-node r is labelled by $\neg\varphi$ and whose construction is continued by means of the following decomposition rules.

Table 1: Classical Decomposition Rules

Given a node m on a branch, labelled by				
$\varphi \wedge \psi$	$\neg(\varphi \wedge \psi)$	$\varphi \vee \psi$	$\neg(\varphi \vee \psi)$	$\neg\neg\varphi$
we can add to that branch				
two	two	two	two	one node n
further nodes n_1, n_2 with the branch				
not forking	forking	forking	not forking	
where n_1, n_2 are labelled respectively by the formulae				labelled by
φ, ψ	$\neg\varphi, \neg\psi$	φ, ψ	$\neg\varphi, \neg\psi$	φ

All these decomposition rules act on a single input. The number of formulae in the output may be one (rule for negation) or two (all the others) and, in in the latter case, with or without forking, according to the rule under consideration. Decomposition always terminates, on the understanding that no step is repeated on any given branch.

A completed tree is said to be acceptable iff every branch contains a crash-pair, i.e. a pair of nodes $c_1 : \gamma,\ c_2 : \neg\gamma$ for some formula γ. Acceptability is independent of the order in which decomposition is carried out (although that can influence the shape and size of the tree). It is known that the classical tautologies are just those formulae that have an acceptable decomposition tree, i.e. such that there is an acceptable decomposition tree with root labelled by $\neg\varphi$.

3 Decomposing Arrows

We add decomposition rules for the implication connective in both plain and negated occurrences.

Table 2: Decomposition Rules for Arrow

modus ponens	**counter-case**
Given nodes m, m' on a branch labelled by	Given node m on a branch labelled by
$\varphi, \varphi \to \psi$	$\neg(\varphi \to \psi)$
we can add to that branch without forking	
a further node n, labelled by	two nodes n_1, n_2, labelled by
ψ	$\varphi,\ \neg\psi$

The rule for decomposing unnegated arrows, modus ponens, differs from that used in previous decomposition/tableau systems for relevance logic. In all those of which the

present author is aware (see appendix 3), a rule of 'implicative forking' is used, the branch forking with the negation of the antecedent on one arm and the consequent on the other, echoing what is done in classical logic for material implication when it is taken as primitive. The reason for this change will be explained shortly, in section 4.

No constraints are placed on our two rules at their points of application; but a global constraint concerning counter-case will feature in the definition of a directly acceptable decomposition tree. Before giving that definition, we make some general remarks on the use of *reductio ad absurdum* in the context of relevance logic, and the rationale behind our rule of counter-case.

Evidently, any construction carried out in terms of semantic decomposition trees, where we label the root with the negation of the target formula and work towards explicit contradictions in every branch, makes use of a form of *reductio ad absurdum*. It may be imagined from the rejection of the classical principle of 'explosion' $(p \wedge \neg p) \rightarrow q$ and the role of the four-valued de Morgan algebra in some semantic constructions for relevance logics, that their spirit is intrinsically hostile to *reductio*. But this is not the case. Indeed, in the standard natural deduction presentation of the relevance logic R, if we can relevantly derive each of $\gamma, \neg\gamma$ from $\neg\theta$, using that formula and it alone as assumption in both derivations, then we may relevantly derive $\gamma \wedge \neg\gamma$ from $\neg\theta$ and thus derive $\neg\theta \rightarrow (\gamma \wedge \neg\gamma)$ from the empty premise set. Contraposing and de Morganizing gives us $(\gamma \vee \neg\gamma) \rightarrow \theta$ from the empty set, so detaching using the provable formula $\gamma \vee \neg\gamma$ produces θ. Thus, there is nothing incoherent about the idea of allowing *reductio* a central role in an account of relevance logic.

The counter-case rule differs conceptually from modus ponens, as also from the decomposition rules for the truth-functional connectives in Table 1. For them, the input relevantly implies each element of the output (or, for the forking rules, their disjunction). The same would hold of counter-case under a reading of $\rightarrow$ as material implication. But when $\rightarrow$ is read as relevant implication then, intuitively, $\neg(\varphi \rightarrow \psi)$ does not imply either of the output elements $\varphi, \neg\psi$, less so both. The rationale for the counter-case rule is indirect: the formulae $\varphi, \neg\psi$ may be thought of as *wlog* (without loss of generality) assumptions. The situation is reminiscent of that which arises when we decompose an existential quantification in a tree for classical first-order logic, passing from $\exists x(\varphi)$ to $\varphi_{x:=a}$ where a is a 'fresh' constant, that is, one that does not occur anywhere in the branch up to that point. Clearly, the conclusion of that decomposition is not in general a consequence of its premise (although they are friendly in the sense defined in Makinson 2007). But whilst $\exists x(\varphi) \not\models \varphi_{x:=a}$ we do have that $A, \exists x(\varphi) \models \gamma \wedge \neg\gamma$ whenever $A, \varphi_{x:=a} \models \gamma \wedge \neg\gamma$ and a does not occur in any formula in $A \cup \{\varphi\}$, so that the decomposition procedure is sound.

4 Directly Acceptable Decomposition Trees and Formulae

In this section, we define directly acceptability for trees and formulae; in section 8 we will extend it recursively to a notion of acceptability *tout court*. For present purposes, we need the notions of a *crash-pair*, *trace* and *critical node*, leading to the central *parity constraint*.

Crash-pairs are defined as in the classical context: they are pairs of nodes $c_1 : \gamma$, $c_2 : \neg\gamma$ for some formula γ. The notion of *trace* is also a natural extension of its classical counterpart. Conceptually, the trace of a node n in a decomposition tree is the set of all nodes in the tree that are used in getting to it. Technically, it is the least set of nodes of the tree that contains n, and is such that whenever it contains a non-root node then it also contains the node (both nodes, for modus ponens) from which it was obtained by application of a decomposition rule. The trace of a set of nodes is understood to be the union of their separate traces.

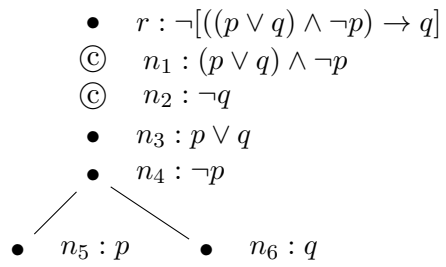
The notions of critical pair and the parity constraint are specific to the relevantist context. A *critical pair* is a pair $\{n_1 : \varphi, n_2 : \neg\psi\}$ of nodes that are introduced by an application of the counter-case rule to a node $m : \neg(\varphi \to \psi)$; the individual nodes n_1, n_2 are called *critical nodes* and are *partners* of each other.

A crash-pair $C = \{c_1 : \gamma, c_2 : \neg\gamma\}$ is said to satisfy the *parity constraint* iff for every critical node n, if n is in the trace of C then so too is its partner. In other words, such that for every critical pair of nodes in the tree, either both of them are in the trace of C, or neither of them are. A branch B of a decomposition tree is said to *crash with parity* iff it contains some crash-pair $C_B = \{c_1 : \gamma, c_2 : \neg\gamma\}$ that satisfies the parity constraint.

Finally, a *directly acceptable decomposition tree* is one such that every branch B crashes with parity. A formula is called *directly acceptable* iff it has some directly acceptable tree, i.e. such that there is an acceptable decomposition tree with root labelled by $\neg\varphi$.

The following remarks may help appreciate the finer contours of these definitions.

(1) The requirement of crashing with parity is reminiscent of the proviso, in natural deduction systems for relevance logic, that suppositions should be used in derivations. This is not surprising, since the intuitive rationale for the decomposition rule for negated arrows, described above, treats the critical nodes as suppositions of a *reductio ad absurdum*. The effect of the constraint can be illustrated by disjunctive syllogism $((p \vee q) \wedge \neg p) \to q$, which is relevantistically objectionable because it so easily yields 'right explosion' $(p \wedge \neg p) \to q$. If we construct a decomposition tree for disjunctive syllogism we naturally get the following.



Here and throughout the paper non-critical nodes are diagrammed by • and critical ones by ©. Both branches of the tree crash, but the left branch does not do so with parity, since the critical node n_1 is in the trace of the unique crash-pair $\{n_4, n_5\}$ on that branch but its partner n_2 is not.

(2) It is the parity requirement that leads us to decompose unnegated arrows by modus ponens rather than by implicative forking as in classical logic. This is illustrated in Appendix 1 when decomposing axiom schemes for R, with details spelled out in the simplest instance, the scheme of assertion.

(3) The quantificational structure of the definition of direct acceptability is quite complex: $\forall B \exists C \forall c$, where the variables range over branches, crash-pairs and critical nodes respectively. Its delicacy can be illustrated by trees for symmetric explosion and mingle (given after Observation 2 below) that are not directly acceptable but do satisfy the weaker $\forall B \forall c \exists C$ condition.

(4) Clearly the set of directly acceptable formulae is decidable. Given the undecidability of R (Urquhart 1984), this already tells us that the two cannot coincide.

(5) Some further aspects of the definitions are reviewed in the appendices. A normal form for our decomposition trees is given at the end of Appendix 1. The treatment of disjunction is compared, in Appendix 2, with the way it is handled in standard systems of natural deduction for relevance logic. The relationship of our trees to earlier tree/tableau approaches to relevance logic is reviewed in Appendix 3.

5 Direct Acceptability – Exclusions

This section establishes some limits to direct acceptability, that is, results of the kind 'only formulae with such-and-such a property are directly acceptable'.

Observation 1. *When $\rightarrow$ (as well as $\wedge, \vee, \neg$) is read truth-functionally as material implication, every directly acceptable formula is a tautology.*

Proof. Suppose that φ is directly acceptable. Then it has a directly acceptable decomposition tree with root $r : \neg\varphi$; choose any one of them, T. Every branch of T contains a

crash-pair. Since both *modus ponens* and counter-case rule are sound for truth-functional implication (and all other decomposition rules are known to be sound for their connectives), it follows that $\neg\varphi$ is classically unsatisfiable, so that φ is a tautology. □

Observation 2. *Direct acceptability satisfies the letter-sharing condition. That is, when φ, ψ share no sentence letters, then $\varphi \to \psi$ is not directly acceptable.*

The Observation follows from a rather elaborate but useful lemma.

Lemma for Observation 2. *Let φ, ψ be formulae with no sentence letters in common. Then for every directly acceptable decomposition tree T for $\varphi \to \psi$, every node $m : \theta$ in T after the root $r : \neg(\varphi \to \psi)$ has in its trace exactly one of the critical nodes $n_1 : \varphi$ and $n_2 : \neg\psi$ obtained from the root by the counter-case rule, and every sentence letter occurring in θ occurs in the formula, φ or $\neg\psi$, attached to that node.*

Proof. We induce on the construction of T from its root. The base concerns the cases that $m : \theta = n_1 : \varphi$ or $m : \theta = n_2 : \neg\psi$, since these are the only items immediately obtainable from a root of the form $r : \neg(\varphi \to \psi)$ by a single application of the decomposition rules available, and the property is immediate. For the induction step, there are two cases to consider.

Case 1. Suppose that m is obtained from a single node m' other than the root by one of the decomposition rules other than *modus ponens*. By the induction hypothesis, m' has the property in question. Since $\text{trace}(m) = \text{trace}(m') \cup \{m\}$ it follows that m also has in its trace exactly one of the critical nodes $n_1 : \varphi$ and $n_2 : \neg\psi$ obtained from the root by counter-case; and since the decomposition rules do not introduce any new sentence letters, m also satisfies the second part of the property.

Case 2. Suppose that $m : \theta$ is obtained from a pair of nodes $m' : \alpha$ and $m'' : \alpha \to \theta$ by modus ponens. Neither of m', m'' can be the root-node – the latter because of the form of $\alpha \to \theta$ and the former because is shorter than $\alpha \to \theta$ and so shorter than the root node formula. So by the induction hypothesis, exactly one of $n_1 : \varphi$ and $n_2 : \neg\psi$ is in the trace of $m' : \alpha$, and likewise for $m'' : \alpha \to \theta$. This gives rise to four subcases, coming in two similar pairs.

Subcase 2.1.1. Suppose that $n_1 : \varphi$ is in the trace of both $m' : \alpha$ and $m'' : \alpha \to \theta$. Then $n_2 : \neg\psi$ is in the trace of neither of those two nodes. Hence $n_1 : \varphi$ is in the trace of $m : \theta$ while $n_2 : \neg\psi$ is not, so that exactly one of the two is in the trace of $m : \theta$. Moreover, every letter in θ occurs in $\alpha \to \theta$, and so by the induction hypothesis occurs in $n_1 : \varphi$, as desired.

Subcase 2.1.2. Suppose that $n_2 : \neg\psi$ is in the trace of both of $m' : \alpha$ and $m'' : \alpha \to \theta$. The argument is similar to that for subcase 2.1.1.

Subcase 2.2.1. Suppose that $n_1 : \varphi$ is in the trace of $m' : \alpha$ while $n_2 : \neg\psi$ is in the trace of $m'' : \alpha \to \theta$. We derive a contradiction. By the induction hypothesis, every letter in α occurs in φ and every letter in $\alpha \to \theta$ occurs in $\neg\psi$. From the latter, every letter in α occurs in ψ. Since α contains at least one letter, this implies that φ and ψ share a letter, contrary to the initial supposition of the Lemma.

Subcase 2.2.2. Suppose that $n_1 : \varphi$ is in the trace of $m'' : \alpha \to \theta$ while $n_2 : \neg\psi$ is in the trace of $m' : \alpha$. The argument is similar to that for subcase 2.2.1. □

Proof of Observation 2 from the Lemma. Suppose for reductio that T is a directly acceptable decomposition tree with root $r : \neg(\varphi \to \psi)$, and that φ shares no sentence letters with ψ. Let B be any branch of T, with crash-pair $C_B = \{c : \gamma, c' : \neg\gamma\}$. By the Lemma, exactly one of $n_1 : \varphi$ and $n_2 : \neg\psi$ is in the trace of c, and exactly one of them is in the trace of c'. If n_1 (resp. n_2) is in the trace of both of c, c', then n_2 (resp. n_1) is in the trace of neither of c, c', violating the parity condition. Thus n_1 is in the trace of c while n_2 is in the trace of c', or inversely. Consider the first case, the second is similar. By the Lemma, every letter in γ occurs in φ and every letter in $\neg\gamma$ occurs in ψ. Since γ has at least one letter, it follows that the formulae φ, ψ share at least one letter, contradicting the initial supposition. □

To illustrate Observation 2, it is instructive to consider the formula $(p \wedge \neg p) \to (q \vee \neg q)$ (symmetric explosion), which notoriously fails the letter-sharing condition. The Observation tells us that it is not directly acceptable. This may be puzzling, since the following tree for it may at first sight seem to be so.

- $r : \neg[(p \wedge \neg p) \to (q \vee \neg q)]$
- ⓒ $n_1 : p \wedge \neg p$
- ⓒ $n_2 : \neg(q \vee \neg q)$
- $n_3 : p$
- $n_4 : \neg p$
- $n_5 : \neg q$
- $n_6 : \neg\neg q$

The tree has a unique branch, with a single critical pair $\{n_1, n_2\}$ and two crash-pairs $\{n_3, n_4\}$ and $\{n_5, n_6\}$. But $\{n_3, n_4\}$ fails the parity constraint because n_1 is in its trace while its partner n_2 is not and, similarly, $\{n_5, n_6\}$ fails the constraint because n_2 is in its trace while n_1 is not. This illustrates the importance of the order of the quantifiers $\forall B \exists C \forall c$ in the definition of a directly acceptable tree, contrasting with $\forall B \forall c \exists C$; for symmetric explosion, the latter is satisfied since for every critical node c there is a crash-pair C such that (vacuously) if c is in the trace of C then so is its partner.

A similar pattern emerges in the decomposition tree for $\neg(p \to p) \to (q \to q)$, which we might call 'symmetric explosion for arrow'. It also appears for mingle, $p \to (p \to p)$ which,

unlike symmetric explosion, satisfies the letter-sharing condition. For mingle, we have the following decomposition tree.

$$\bullet\ r : \neg[p \to (p \to p)]$$
$$\copyright\ n_1 : p$$
$$\copyright\ n_2 : \neg(p \to p)$$
$$\copyright\ n_3 : p$$
$$\copyright\ n_4 : \neg p$$

There is a single branch, with two crash-pairs $\{n_1, n_4\}$ and $\{n_3, n_4\}$ sharing the node n_4. But the critical node n_3 is not in the trace of the first crash-pair although its partner n_4 is, while the critical node n_1 is not in the trace of the second one although its partner n_2 is. Hence the tree is not directly acceptable, failing the $\forall B \exists C \forall c$ condition although it satisfies the weaker $\forall B \forall c \exists C$ one.

Mingle and symmetric explosion thus get 'quite close' to direct acceptability, missing out only on the order of the quantifier prefix in the parity condition. This formal fact resonates with a vague but strong intuition that they are, in some sense, among the least irrelevant of the formulae rejected by relevantists. In contrast, the formula $p \to (q \to p)$, of which mingle is a substitution instance, and which we call 'mangle' in preference to the more common but rather bland name 'positive paradox', is intuitively irrelevant to a more serious extent, corresponding to the fact that its tree (write q in place of p at appropriate places in the above) fails even the weaker $\forall B \forall c \exists C$ condition.

We recall also that there are links between these 'least irrelevant' formulae. Meyer has shown that the system RM defined by adding mingle to the standard axiomatization of R contains $\alpha \to \beta$ whenever it contains both $\neg\alpha$ and β. In particular, RM contains both of the above forms of symmetric explosion, derivations of which can also be found in Anderson & Belnap 1975 section 29.5 and Priest 2008 section 10.11, question 6.

6 Direct Acceptability – Inclusions

We now show the surprisingly broad reach of direct acceptability, with results of the kind 'such-and-such formulae are directly acceptable' and 'the set is closed under such-and-such operations'.

Observation 3. *All axiom schemes of the standard axiomatization of the relevance logic R are directly acceptable.*

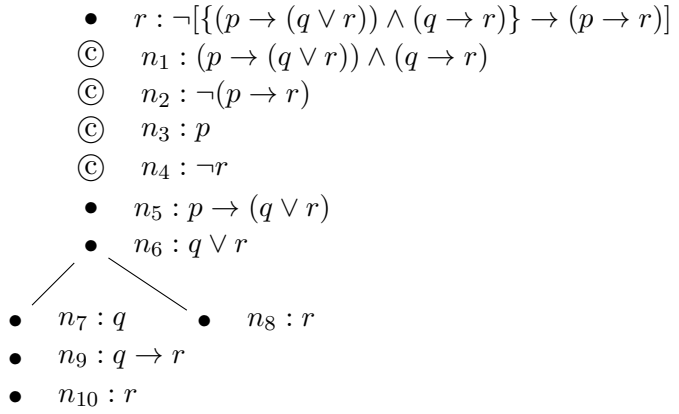
It suffices to check off the axiom schemes one by one, finding a directly acceptable decomposition tree for each. The verifications are routine but often interesting, and are given in Appendix 1.

Observation 4. *The set of all directly acceptable formulae is closed under substitution and conjunction.*

Proof. For conjunction, simply put the trees together with a new root to which is applied the decomposition rule for negated conjunctions. For substitution, substitute throughout the tree; if the original tree is directly acceptable then so is the one obtained. □

Observation 5. *There are directly acceptable formulae that are not in R.*

Proof. The formula $\{(p \to (q \vee r)) \wedge (q \to r)\} \to (p \to r)$ is known to be absent from R (using the matrix M_0 of Anderson & Belnap 1975, pp. 252-253), but has the following directly acceptable decomposition tree.



□

We remark in passing that this formula, given in Dunn 1986 section 4.6, is the simplest of several that were shown by Dunn, Meyer and Urquhart not to be theses of R while valid in Urquhart's semi-lattice semantics; two others are $\{(p \to (q \vee r)) \wedge (q \to s)\} \to (p \to (s \vee r))$ and $[(p \to p) \wedge ((p \wedge q) \to r) \wedge (p \to (q \vee r))] \to (p \to r)$ (see Humberstone 2011 page 1210, example 8.13.21, Bimbó & Dunn 2017 section 4). All three conspicuously involve disjunction alongside the arrow, with discrete assistance from conjunction. One may ask how they should be seen from a relevantist perspective. Our view is that bare intuition gives us little guidance on the question and may legitimately be educated by the outcome of a satisfying formal account.

7 Direct Acceptability – A Limitation

However, the example illustrating Observation 5 also points to a limitation of direct acceptability. Clearly, in R that formula is inter-derivable, using contraposition and then commutation in the antecedent, with the formula $\{(p \to q) \wedge ((p \wedge q) \to r)\} \to (p \to r)$

expressing cumulative transitivity of the arrow, so that the latter is likewise absent from R. However, unlike $\{(p \to (q \vee r)) \wedge (q \to r)\} \to (p \to r)$, it is not directly acceptable, as is easily checked. This suggests that there is something incomplete about direct acceptability, which the following observation makes manifest.

Observation 6. *The set of all directly acceptable formulae is not closed under detachment.*

To avoid any misunderstanding, we emphasize that the question of the closure of the set of all directly acceptable formulae under detachment is quite different from the presence of modus ponens as one of the decomposition rules that is allowed when building trees. Observation 6 tells us, in effect, that the latter does not imply the former. It is thus advisable to retain different terms for the two operations.

Proof of Observation 6. Let φ be the formula of Observation 5, and ψ the formula expressing cumulative transitivity. Then each of $\varphi, \varphi \to \psi$ is directly acceptable, but ψ is not. □

Although this one example suffices to establish Observation 6, we mention several more, because their variety casts light on the reasons for the failure and helps guide the way to an appropriate response. In each example, we specify formulae ψ, φ with ψ not directly acceptable although $\varphi, \varphi \to \psi$ are so. The verifications are routine and so omitted but, in the author's experience, a lot can be learned by walking through them.

An example with just arrow as connective was noted by Lloyd Humberstone in correspondence with the author. It puts $\psi = ((p \to p) \to q) \to q$ and $\varphi = p \to p$. In both this example and that used in the proof of Observation 6 above, the formula ψ would become directly acceptable if we added modus tollens alongside modus ponens as a rule for decomposing unnegated arrows.

However, rescue by modus tollens is not always possible, for instance when $\psi = \neg[(p \vee \neg p) \to (p \wedge \neg p)]$ and $\varphi = (p \vee \neg p) \wedge \neg(p \wedge \neg p)$. Here ψ is a negated conditional and, quite generally, no negated conditional has a directly acceptable tree, even if modus tollens supplements modus ponens. For, given a root labelled $\neg\neg(\alpha \to \beta)$ the only decomposition rule we can apply is double negation elimination to get $\alpha \to \beta$, and the only rule that might then be applied to $\alpha \to \beta$ is modus ponens (or tollens), but there is no minor premise available to do so.

Another example where addition of modus tollens is still of no avail, puts $\psi = (p \to \neg p) \to \neg(\neg p \to p)$ and $\varphi = (p \vee \neg p)$. Here, φ and $\varphi \to \psi$ are directly acceptable (noting that to ensure crash with parity for $\varphi \to \psi$, we need to apply modus ponens twice in each branch, not stopping at the first crash-pair found), but ψ is not directly acceptable (even with the help of modus tollens).

On the other hand, in the last two examples, ψ does become directly acceptable if we add implicative forking as a further rule for decomposing unnegated arrows. But there are also quite simple examples where that does not suffice to rescue ψ, even when used in

combination with modus tollens and ponens. Such is the case for $\psi = (\neg\neg p \to \neg\neg q) \to (p \to q)$ and $\varphi = p \to \neg\neg p$, likewise for $\psi = ((p \wedge p) \to (q \wedge q)) \to (p \to q)$ and $\varphi = p \to (p \wedge p)$. In both instances, a 'relevantistically simplifiable' formula is buried inside ψ in a way that is not accessible by applications of modus ponens or tollens in the decomposition tree; and while it can be accessed by implicative forking, doing so creates branches that violate the crash-with-parity requirement.

In each of our six examples, while the consequent ψ is not directly acceptable, it nevertheless appears to be agreeable from a relevantist point of view; indeed, except for the very first, they are theses of R. The question thus arises: Is there a way of extending our decomposition procedure to validate these formulae and, more generally, to ensure closure under detachment – without losing the vital letter-sharing property?

8 Going Recursive

To answer that question, it is natural to introduce a recursive step into the construction of decomposition trees. The basis of the recursion takes directly acceptable trees (as defined in section 4) to be acceptable. For the recursion step, given an acceptable tree for a formula $\varphi \to \psi$, we allow passage from a node $m : \varphi$ to a node $n : \psi$ in the construction of further acceptable trees.

Strictly speaking, the recursive step is not one of decomposition, since there is no limit on the complexity of the formula ψ compared to φ, nor even on the choice of its sentence letters. Thus, with a formula called *acceptable* iff it has some acceptable tree, decidability may well be lost although semi-decidability, i.e. recursive enumerability, is clearly retained.

Acceptability has some very attractive features.

Observation 7. *All directly acceptable formulae are acceptable. Modus tollens and implicative forking are admissible as decomposition rules in the construction of acceptable decomposition trees. The set of acceptable formulae is closed under substitution, conjunction and detachment.*

Proof. Modus tollens is admissible, since it may be taken as abbreviating an application of the recursive rule using the directly acceptable formula $(\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$, followed by modus ponens. Similarly for implicative forking, using the directly acceptable $(\alpha \to \beta) \to (\neg\alpha \vee \beta)$ followed by the rule for decomposing disjunctions. Closure under substitution and conjunction are checked in the same way as in the context of direct acceptability, while detachment is verified as follows. Suppose α and $\alpha \to \beta$ are both acceptable. From the latter, $\neg\beta \to \neg\alpha$ is acceptable, since we can build a tree with its negation as root, apply counter-case to get $\neg\beta$ and $\neg\neg\alpha$, double negation elimination to get α, then use the acceptable formula $\alpha \to \beta$ in the recursive rule to get β; this gives us a crash-pair with both critical nodes in its trace. Finally, it follows that β is also acceptable: build a tree with its negation as root, use the acceptable formula $\neg\beta \to \neg\alpha$ in the recursive rule to get

$\neg\alpha$, then paste in the acceptable tree for α. Clearly, since every branch of the tree for α crashes with parity, so too for this one. □

Corollary. *All theses of the system R are acceptable (but not conversely).*

Proof. Proof. Immediate from Observations 3, 7, 5. □

We note that all our examples of formulae illustrating Observation 6, while not directly acceptable are nevertheless acceptable, as can be verified directly (or via Observation 7, or again for all but the first example, by the Corollary). The two formulae of Meyer, Dunn, and Urquhart mentioned in the last paragraph of section 6, are also acceptable, the first directly so.

On the other hand, we have not been able to determine whether the letter-sharing property continues to hold when the recursive rule is introduced. This is the open question of the title. As letter-sharing is an essential feature for any relevance logic worthy of the name, the entire procedure of this paper collapses if the answer is negative. If, on the other hand, the answer is positive, then our procedure would appear to supply a clear rationale for relevance logic. Note that by Observation 7, its Corollary, and the remark in the last paragraph of section 5, a positive answer to the question would also show that mingle and mangle are inacceptable

One could also obtain all the positive features of Observation 7 simply by closing the set of directly acceptable formulae under detachment. In effect, this is to treat that (decidable) set as constituting the axioms of a Hilbertian system, with detachment as the sole derivation rule. It is immediate from Observation 7 that all the derivable formulae of such a system are acceptable in the sense we have defined; but we do not know whether the converse holds, nor whether the set of derivable formulae of such a Hilbertian system has the letter-sharing property.

We have not followed this alternative option, for two reasons. While the recursive rule appears conceptually natural as a recycling device in the construction of trees, simple closure of the set of directly acceptable formulae under detachment has the air of an unprincipled add-on. It also takes attention away from decomposition trees to Hilbertian derivations, with decomposition serving only to generate the axioms.

Appendix 1: Verifications for Observation 3

Observation 3. *All axiom schemes of the relevance logic R are directly acceptable.*

We are referring to the standard axiomatization in Anderson et al. 1992 page xxiv (also in Mares 2004 Appendix A), which uses the basic connectives $\neg, \wedge, \vee, \rightarrow$. Some formulations of R in the literature add auxiliary primitives, notably a two-place connective $\circ$ of 'fusion'

and/or a zero-ary connective (propositional constant) $\mathbf{t}$, with a view to facilitating investigations of the basic system. Our decompositional account has no need for the auxiliary apparatus.

Verification. For the first-degree axiom schemes, that is, those of the form $\alpha \to \beta$ where neither antecedent or consequent contains arrows, one can simply use the natural classical decomposition tree for $\alpha \supset \beta$, beginning with counter-case to get $\alpha, \neg\beta$ and then applying the rules for the classical connectives $\neg, \wedge, \vee$ only, and checking by inspection that every branch satisfies parity. That covers axiom schemes 1 (identity) $\alpha \to \alpha$; 5 ($\wedge$-elimination) $(\alpha \wedge \beta) \to \alpha$ and $(\alpha \wedge \beta) \to \beta$; 6 ($\vee$-introduction) $\alpha \to (\alpha \vee \beta)$ and $\beta \to (\alpha \vee \beta)$; 9 (distribution) $(\alpha \wedge (\beta \vee \gamma)) \to ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$; and 11 (double negation elimination) $\neg\neg\alpha \to \alpha$. For each of the remaining schemes, which are of higher degree, we exhibit a directly acceptable decomposition tree with comment.

Scheme 2 (suffixing): $(\alpha \to \beta) \to ((\beta \to \gamma) \to (\alpha \to \gamma))$. The unique branch contains a unique crash-pair with all six critical nodes in its trace, so parity is satisfied. The linearity of the tree (i.e. single branch) is a consequence of the fact that the only connectives involved are $\to, \neg$ (likewise for assertion and contraction below).

- $r : \neg[(\alpha \to \beta) \to ((\beta \to \gamma) \to (\alpha \to \gamma))]$
- ⓒ $n_1 : \alpha \to \beta$
- ⓒ $n_2 : \neg[(\beta \to \gamma) \to (\alpha \to \gamma)]$
- ⓒ $n_3 : \beta \to \gamma$
- ⓒ $n_4 : \neg(\alpha \to \gamma)$
- ⓒ $n_5 : \alpha$
- ⓒ $n_6 : \neg\gamma$
- •$n_7 : \beta$
- •$n_8 : \gamma$

Scheme 3 (assertion): $\alpha \to ((\alpha \to \beta) \to \beta)$. This is the only axiom scheme from the list that is unprovable in the weaker systems NR and E, which are intended to capture a composite notion of relevant-and-necessary implication (see e.g. section 28.1 of Anderson & Belnap 1975 or section 4 of Mares 2012). Assertion is directly acceptable: the unique branch contains a unique crash-pair, and all four critical nodes are in its trace.

• $r : \neg[\alpha \to ((\alpha \to \beta) \to \beta)]$
Ⓒ $n_1 : \alpha$
Ⓒ $n_2 : \neg[(\alpha \to \beta) \to \beta)]$
Ⓒ $n_3 : \neg\beta$
Ⓒ $n_4 : \alpha \to \beta$
• β

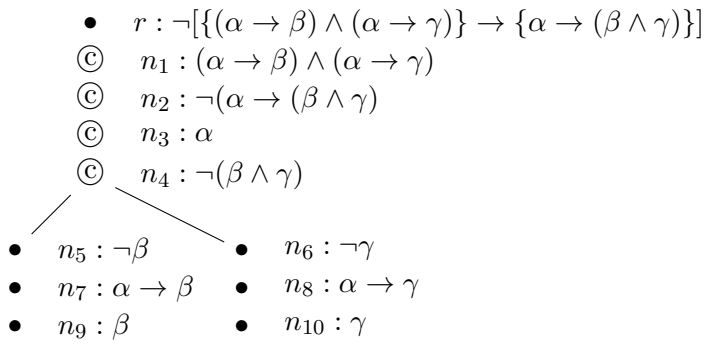
Comment: If we were to handle unnegated arrows by the classical decomposition rule of implicative forking, then parity would fail for all the higher-degree axiom schemes of R. This is most concisely illustrated with assertion, since its tree has only one unnegated arrow, at node n_4. Imagine that from n_4 we were to fork left to $\neg\alpha$ and right to β. Then we would have a crash-pair $\alpha, \neg\alpha$ on the left branch, but critical node n_3 would not be in its trace although its partner n_4 is. Moreover, we would have a crash-pair $\beta, \neg\beta$ on the right branch, but critical node n_1 would not be in its trace, although its partner n_2 is. In other words, in this example forking for the conditional obviates any need on the left branch for the negation of its consequent; and dispenses with any call on the right branch to the antecedent of a previous negated conditional.

Scheme 4 (contraction): $(\alpha \to (\alpha \to \beta)) \to (\alpha \to \beta)$. The unique branch contains a crash-pair $\{n_4, n_6\}$ with all four critical nodes in its trace.

• $r : \neg[(\alpha \to (\alpha \to \beta)) \to (\alpha \to \beta)]$
Ⓒ $n_1 : \alpha \to (\alpha \to \beta)$
Ⓒ $n_2 : \neg(\alpha \to \beta)$
Ⓒ $n_3 : \alpha$
Ⓒ $n_4 : \neg\beta$
• $n_5 : \alpha \to \beta$
• $n_6 : \beta$

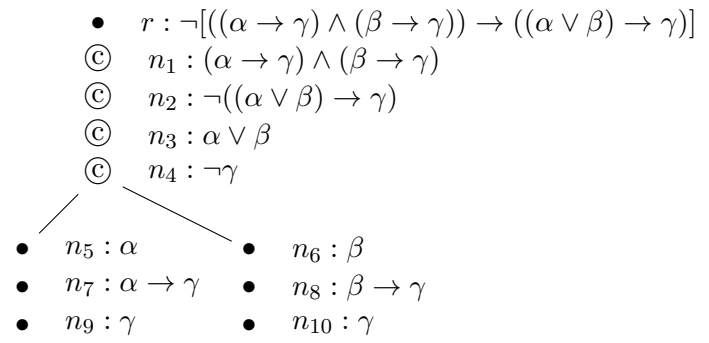
Comment: An interesting feature of this tree is that it also contains the crash-pair $\{n_2, n_5\}$ which, however, does not satisfy the parity condition since the critical node n_4 is not in its trace although its partner n_3 is.

Scheme 7 ($\wedge$-introduction) $\{(\alpha \to \beta) \wedge (\alpha \to \gamma)\} \to \{\alpha \to (\beta \wedge \gamma)\}$. Two branches each with its crash-pair, every critical node in the trace of each crash-pair.



Comment: In this tree, n_1 is a single critical node labelled by a conjunctive formula. If instead we had two distinct critical nodes labelled by the respective conjuncts, as would be the case if we were decomposing the third-degree formula $(\alpha \to \beta) \to \{(\alpha \to \gamma) \to (\alpha \to (\beta \wedge \gamma))\}$, then both branches would still crash, but without parity. This illustrates a general point. From a relevantist point of view, there is a significant difference between classically equivalent formulae $(\varphi \wedge \psi) \to \theta$ and $\varphi \to (\psi \to \theta)$. In terms of trees, this is because decomposing $\varphi \to (\psi \to \theta)$ gives rise to an additional application of counter-case, thus a further critical pair that can fail the parity condition. Remarkably however, suffixing (scheme 2 above), which is a form of transitivity, remains acceptable even in its third-degree version. On the other hand, cumulative transitivity fares less well: its second-degree form $\{(p \to q) \wedge ((p \wedge q) \to r)\} \to (p \to r)$ is acceptable (though not directly so, as noted in Observation 6), but its third-degree version $(p \to q) \to \{((p \wedge q) \to r) \to (p \to r)\}$ does not appear to be so.

Scheme 8 ($\vee$-elimination): $((\alpha \to \gamma) \wedge (\beta \to \gamma)) \to ((\alpha \vee \beta) \to \gamma)$. Same comments as for scheme $\wedge+$.



Scheme 10 (contraposition): $(\alpha \to \neg\beta) \to (\beta \to \neg\alpha)$. The unique branch ends in a crash-pair, and each of the four critical nodes is in its trace.

- $r : \neg[(\alpha \to \neg\beta) \to (\beta \to \neg\alpha)]$
- ⓒ $n_1 : \alpha \to \neg\beta$
- ⓒ $n_2 : \neg(\beta \to \neg\alpha)$
- ⓒ $n_3 : \beta$
- ⓒ $n_4 : \neg\neg\alpha$
- $n_5 : \alpha$
- $n_6 : \neg\beta$

The decomposition trees that we have use in the verification of Observation 3 all display a certain normal form: (1) Each branch stops at its designated crash-pair; (2) In every application of counter-case, the two output critical nodes are contiguous; (3) the designated crash-pair of each branch has in its trace *both* of the critical nodes of every application of counter-case carried out in that branch (which is more than the both-or-none required by parity). It is not difficult to show that such a form is always available, that is, if a formula has a directly acceptable (resp. acceptable) decomposition tree, then it has one satisfying those three conditions.

Appendix 2: Disjunction

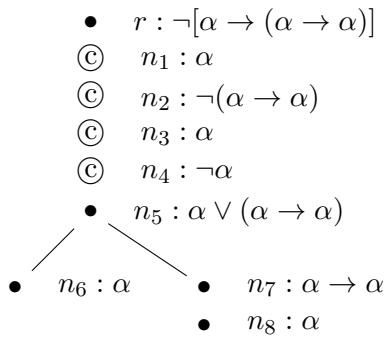
This appendix compares our treatment of disjunction with the way it is handled in the standard system of natural deduction for the relevance logic R (Anderson & Belnap 1975).

To fix ideas, we begin with the example of the formula $\{(p \to (q \vee r)) \wedge (q \to r)\} \to (p \to r)$ which, as noted in Observation 5, is directly acceptable but not in R. It is instructive to recall how an attempt to derive it using natural deduction for R fails its proviso on $\vee-$. From the suppositions $(p \to (q \vee r)) \wedge (q \to r)$ and p one reaches $q \vee r$; classically one would then make two sub-proofs, the first supposing q to get r and the second supposing r to get r. But while the first sub-derivation appeals to the supposition $(p \to (q \vee r)) \wedge (q \to r)$, the second does not, violating the proviso that the two sub-derivations must make use of the same suppositions (other than q and r themselves). On the other hand, in our decomposition procedure the critical node labelled by $(p \to (q \vee r)) \wedge (q \to r)$ is used in decomposing to $q \vee r$ and, while the tree then forks, the parity constraint acts on each branch in its entirety, not as a comparison of sub-branches.

So, one may ask, why don't mangle $\alpha \to (\beta \to \alpha)$ and its substitution instance mingle $\alpha \to (\alpha \to \alpha)$ also come out as directly acceptable? Why doesn't the decomposition procedure allow us to replicate the notorious classical natural deduction in which one supposes α, then supposes β, uses $\vee+$ to get $\alpha \vee (\beta \to \alpha)$, and then carries out sub-derivations with suppositions α and $\beta \to \alpha$ respectively?

For direct acceptability, the answer is simply that trees can only decompose, never compose, so that they have nothing corresponding to $\vee+$. For the more general notion

of acceptability using the recursive rule of section 8, the answer is little more complex. As remarked there, it is still an open problem to show that mingle is not acceptable; but meanwhile we can see how a natural attempt using the recursive rule, fails.



Up to n_4 this tree is the same as that considered for direct acceptability in section 5, and fails despite the presence of two crash-pairs $\{n_1, n_4\}$ and $\{n_3, n_4\}$, since neither of them satisfies parity. If one continues by applying the recursion rule to n_1 to get n_5 and branching as above, the left branch gets a new crash-pair $\{n_4, n_6\}$, but that also fails parity since the critical node n_3 is not in its trace although its partner n_4 is. Attempts to repair this tree snag. For example, if one gets n_5 from n_3 instead of from n_1 then the critical node n_1 disappears from the trace of $\{n_4, n_6\}$ although its partner n_2 remains.

Thus, formulae such as mangle and mingle, which relevantist natural deduction blocks by its 'same other suppositions' proviso, are excluded from direct acceptability by the parity requirement and natural attempts, like the above, to show them to be acceptable in the recursive sense, snag. On the other hand, the mechanism for direct acceptability allows safe passage for both distribution (see appendix 1) and the rather appealing formula of Observation 5.

Blockage of distribution, forcing it to be postulated independently in a rather ad hoc manner, has for long been a source of dissatisfaction or at least unease for relevantist natural deduction. Already Anderson & Belnap 1975 section 27.2 devised a less restrictive version of $\vee-$, calling it rule $\vee E^s$, and noted that it implies distribution; but they refrained from recommending it, presumably fearing that it might yield too much. Similar proposals with varied presentations and perhaps differing a little in content were made by Urquhart 1989 (see also the brief remark in his 2016), Dunn & Restall 2002, and Brady 2006. The essential idea behind all of them is that when in a derivation one reaches a formula $\alpha \vee \beta$ and creates auxiliary derivations headed respectively by α, β, those two formulae are not given fresh dependency labels but are taken to depend on the same suppositions as did $\alpha \vee \beta$. This has the effect of enlarging the set of suppositions that are considered as 'used' in each of the auxiliary derivations, thereby cushioning the impact of the 'same suppositions' proviso; both distribution and the formula of Observation 5 become derivable. Note, however, that the move does not get rid of the 'same suppositions' constraint: it still appears as a proviso

on the rules $\wedge+$ and $\vee-$, its force in the latter diminished by the modified definition of dependency.

Appendix 3: Earlier Work on Decomposition Trees for Relevance Logic

Decomposition trees (aka semantic tableaux) for relevance logics were investigated by Dunn 1976 and by McRobbie in his doctoral thesis 1979, with parts of the latter published in the abstracts McRobbie 1977, McRobbie & Belnap 1977 and the paper McRobbie & Belnap 1979. Later work includes Pabion 1979, Bloesch 1993, Priest 2008, Jarmużek & Tkaczyk 2015.

None of these decompose unnegated arrows by modus ponens; instead they fork into the consequent and the negation of the antecedent. Nor do any introduce a recursion into the decomposition procedure. On a more detailed level, we can make the following comparisons.

Dunn's 1976 construction using two trees works beautifully, but covers only first-degree conditionals, i.e. formulae of the form $\varphi \to \psi$ where φ, ψ contain no arrows. It is difficult to extend further.

The language of McRobbie's 1977 abstract lacks negation. It covers only formulae in the positive connectives $\to, \wedge, \vee, \circ$ (non-classical 'fusion') and a truth-constant $\mathbf{t}$, with quite complex decomposition rules. The abstract ends with the remark: "Unfortunately, the problem of extending our result in order to obtain a tableau system for all of R has till this date proved intractable".

Negation is available in McRobbie & Belnap 1977 and 1979, but the language is again severely restricted, for arrow is the only other connective allowed in formulae. Their decomposition trees may still have multiple branches, since unnegated arrows are decomposed by forking. The trees make use of a dependency condition. It is rather loosely expressed but, on the present author's reading, it requires that there is a function taking each branch B to a crash-pair C_B on B such that for every node n there is a branch B containing n such that n is in the trace of C_B.

That condition contrasts with our crash-with-parity constraint, which can likewise be expressed in functional language: *for every critical node n and every branch B, if n is in the trace of C_B then so is its partner*. This monitors only critical nodes, which is less demanding than all nodes, but its second quantifier makes a demand on all branches, rather than just some. The differences are essential for obtaining positive and negative verdicts on distribution and disjunctive syllogism respectively, presumably explaining why McRobbie and Belnap found themselves unable to extend their language to include $\wedge, \vee$.

We should also note a contrast of general perspective. For McRobbie and Belnap, the priority task appears to have been to devise a decomposition procedure whose output coincides with R. As that goal was not reached, a secondary one kicked in: do the same for

some fragment with fewer connectives. Our goal is to articulate transparent decomposition routines whose output set is attractive from a relevantist point of view, with less regard to whether it coincides exactly with the most widely studied relevance logic.

The proposals of Pabion 1979, Bloesch 1993 and Priest 2008 are all inspired by the Routley-Meyer semantics for relevant logics, seeking to make it computationally more manageable by expressing it in terms of decomposition trees. Under the influence of Kripke's earlier work on semantic tableaux for modal logic, they mirror much of the machinery of the Routley-Meyer semantics in the labelling and management of the tableaux themselves – for example, a three-place accessibility relation for tableaux, a range of difficult-to-motivate conditions on the relation, and a star operation for dealing with negation. The proposals thus serve primarily as tableau presentations of the Routley-Meyer semantics rather than independent accounts.

Jarmużek & Tkaczyk 2015 construct decomposition trees for a logic that avoids right explosion, but their goal is very modest. They work only with classical formulae, characterizing in terms of trees the relation that holds between a set A of such formulae and an individual one β iff there is some subset $A' \subseteq A$ such that A' is both classically consistent and tautologically implies β.

Acknowledgements

Heartfelt thanks to Lloyd Humberstone for examples, pointers to literature and wise advice on an early draft; Alasdair Urquhart and Marcello d'Agostino for their comments in correspondence; Michael McRobbie for kindly providing a copy of his dissertation; and students of LSE's PH217/419 (spring 2017) for their questions on a classroom version of the material.

References

Anderson, A.R. & N.D. Belnap Jr, 1975. *Entailment: The Logic of Relevance and Necessity, Volume I.* Princeton: Princeton University Press.

Anderson, A.R., N.D. Belnap Jr & J.M. Dunn, 1992. *Entailment: The Logic of Relevance and Necessity, Volume II.* Princeton: Princeton University Press.

Bimbó, K. & J.M. Dunn, 2017. The emergence of set-theoretical semantics for relevance logics around 1970, *IFCoLog Journal of Logics and their Applications* 4: 557-590.

Bloesch, A. 1993. A tableau style proof system for two paraconsistent logics, *Notre Dame Journal of Formal Logic* 34: 295-301.

Brady, R. 2006. Normalized natural deduction systems for some relevant logics I: the logic DW, *The Journal of Symbolic Logic* 71: 35-66.

Dunn, J.M. 1976. Intuitive semantics for first-degree entailments and 'coupled trees'. *Philosophical Studies* 29: 149-168.

Dunn, J.M. 1986. Relevance logic and entailment. In D. Gabbay and F. Guenthner (eds.) *Handbook of Philosophical Logic*, 1st edn, Vol 3, Dordrecht: Reidel, pp. 117-224.

Dunn, J. M. & G. Restall 2002. Relevance logic. In D. Gabbay and F. Guenthner (eds.) *Handbook of Philosophical Logic*, 2nd edn, Vol. 6, Amsterdam: Kluwer, pp. 1-128.

Humberstone, L. 2011. *The Connectives*. Cambridge, MA: The MIT Press.

Jarmużek, T. & M. Tkaczyk 2015. A method of defining paraconsistent tableaus. In J.-Y.Beziau, M.Chakraborty & S.Dutta (eds.), *New Directions in Paraconsistent Logic*. Berlin: Springer Proceedings in Mathematics & Statistics, vol 152, pp. 295-307.

Makinson, D. 2007. Friendliness and sympathy in logic. In J.-Y. Beziau (ed.) *Logica Universalis*, 2nd edn. Basel: Birkhauser Verlag, pp 191-205.

Mares, E.D. 2012. Relevance logic (updated version of 2012-03-26). *Stanford Encyclopedia of Philosophy* (accessible at https://plato.stanford.edu/entries/logic-relevance/).

McRobbie, M.A. 1977. A tableau system for positive relevant implication (abstract). *Bulletin of the Section of Logic* 6: 131-133, and *Relevance Logic Newsletter* 2: 99-101 (accessible at http://aal.Itumathstats.com/curios/relevance-logic-newsletter).

McRobbie, M.A. 1979. *A proof-theoretic investigation of relevant and modal logics*. Canberra: Australian National University PhD Dissertation.

McRobbie, M.A. & N.D. Belnap Jr 1977. Relevant analytic tableaux (abstract). *Relevance Logic Newsletter* 2: 46-49 (accessible at http://aal.Itumathstats.com/curios/relevance-logic-newsletter).

McRobbie, M.A. & N.D. Belnap Jr 1979. Relevant analytic tableaux. *Studia Logica* 38: 187-200.

Pabion, J.F. 1979. Beth's tableaux for relevant logic. *Notre Dame Journal of Formal Logic* 20: 891-899.

Priest, G. 2008. *An Introduction to Non-Classical Logic: from If to Is*, 2nd edn. Cambridge: Cambridge University Press.

Tennant, N. 1979. Entailment and proofs. *Proceedings of the Aristotelian Society*, New Series 79: 167-189.

Urquhart, A. 1972. Semantics for relevance logics. *The Journal of Symbolic Logic* 37: 159-169.

Urquhart, A. 1984. The undecidability of entailment and relevant implication. *The Journal*

of Symbolic Logic 49: 1059-1073.

Urquhart, A. 1989. What is relevant implication? In J. Norman & R. Sylvan (eds.), *Directions in Relevant Logic*, pp 167-74. Dordrecht: Kluwer.

Urquhart, A. 2016. Relevance logics: problems open and closed. *Australasian Journal of Logic* 13: 11-20.