



Paul Wong*

MINIMISING DISJUNCTIVE INFORMATION

This paper is dedicated to Bob Meyer

Abstract. In [5, 6], Belnap proposed a number of amendments to Rescher’s strategy for reasoning with maximal consistent subsets. More recently in [18], Horty explicitly endorsed Belnap’s amendment to address a related problem in handling inconsistent instructions and commands. In this paper, we’ll examine Belnap’s amendment and point out that Belnap’s suggestion in the use of *conjunctive containment* is open to the very objection he raised. We’ll propose a way out. The strategy turns on the use of *First Degree Entailment* in combination with Quine’s notion of *prime implicate*.

Keywords: paraconsistency, disjunctive information, inconsistencies.

1. Introduction

A common complaint against reasoning based on maximal consistent subsets is that it is too sensitive to the underlying syntax of the logical representation. This may result in information being isolated from each other, and thereby preventing useful information to be extracted. Consider the following example:

Example 1.1. Two information sources may disagree with respect to p while not disagreeing in other respects: $\Gamma = \{p \wedge \neg q, \neg p \wedge (q \vee r)\}$.

*I would like to thank the anonymous referee, participants of the Fourth World Congress on Paraconsistency, in particular, Lloyd Humberstone, Ed Mares, Greg Restall, Graham Priest, Ray Jennings, Peter Schotch, Koji Tanaka, João Marcos and Gillman Payette for their comments and discussions. Lastly, I would like to thank Nuel Belnap for clarification of Conjunctive Containment in a number of email correspondances.

In our example, there is a sufficiently clear sense in which neither $\neg q$ nor $q \vee r$ are directly involved in an inconsistency, though they are conjoined with something that is inconsistent. Splitting Γ into consistent subsets will prevent us from deducing the potentially useful information r . Hence according to Belnap,

[...] Rescher's method gives wildly different accounts depending on just how many ampersands are replaced by commas, or vice versa. It depends too much on how our [...] subtheory [...] is itself separated into sentential bits. [2, p. 544]

Belnap's criticism is fair. It is intuitively implausible that an inference mechanism for handling inconsistent information should give wildly different conclusions for minor syntactic variations in the logical representation. But Belnap's criticism also applies to other formal mechanisms for handling inconsistency such as belief revision. Any *syntax-based* revision of Γ with r would require us to give up at least one member of Γ (see Nebel's [25, 26]). This may incur unwanted information loss.

In [5], Belnap proposed a particular amendment to Rescher's strategy for reasoning with maximal consistent subsets. In [6] Belnap made a further amendment to his earlier amendment. More recently in [18], Horty explicitly endorsed Belnap's second amendment to address a related problem in handling inconsistent instructions and commands. In actual fact, Belnap's suggestions on both occasions amount to the same strategy of finding different ways to *articulate* the input logical representation. According to Belnap, the input logical description Γ is first to be closed under some non-classical logic generating a superset Γ^* and then Rescher's strategy can be applied to Γ^* in the normal way. The role of Γ^* is to make explicit the content of Γ so that the kind of difficulties that arise in situations similar to Example 1.1 can be avoided. In [5], Belnap's suggestion was to use Angell's analytic containment. In [6], Belnap's suggestion was to use an even more restrictive non-classical logic based on the idea of *conjunctive containment*.

In this paper, we'll examine Belnap's amendment and point out that Belnap's suggestion in the use of conjunctive containment is open to the very objection he raised. We'll propose a way out. The strategy turns on the use of *First Degree Entailment* in combination with Quine's notion of *prime implicate*.



2. Rescher Hypothetical Reasoning

In many ways Rescher's method of reasoning from maximal consistent subsets has anticipated many recent developments in AI. One in particular is the nonmonotonic formalism called default logic developed by Reiter in [33]. Since the publication of [33], default logic has been revised and extended (see Schaub [37] for a summary). Many of these new developments have been shown to be expressively equivalent to various forms of belief revision formalism. We'll not be able to summarise all these new developments here. But since many of these extensions are theoretically grounded in some form of reasoning from maximal consistent subsets, Belnap's methodological criticism is still in force here. We'll begin by recalling some standard definitions.

In [34, 35], Rescher proposes a way of reasoning from a set of hypotheses or data which may be inconsistent with a given background theory or a given set of scientific laws. The main idea is that each conclusion we draw from the data should be consistent with our background theory or laws. The strategy is to isolate data that are inconsistent with our background theory. In Rescher's framework, a background theory is first represented in terms of a modal family. Then preferred subsets of data are selected relative to the modal family. And finally conclusions are extracted by applying classical logic to each preferred subset.

DEFINITION 2.1. A modal family \mathcal{M} is a sequence of sets of (propositional) formulae $\langle \Sigma_1, \dots, \Sigma_n \rangle$ such that

1. for each $i < j$, $\Sigma_i \subset \Sigma_j$,
2. for each $i < n$, Σ_i is closed under classical consequence,
3. $\Sigma_n = \Phi$.

A modal family is thus an increasing sequence of classically closed sets such that for each $i < n$, Σ_i is consistent (on pain of violating the proper containment condition). Intuitively, we may think of Σ_1 as the most reliable or confirmed part of a theory and each subsequent Σ_i as less reliable statements of the theory. Our total background theory is then represented by Σ_{n-1} .

DEFINITION 2.2. Relative to a given modal family $\mathcal{M} = \langle \Sigma_1, \dots, \Sigma_n \rangle$ and a set of hypothesis Γ , we define the set of preferred subsets, \mathbb{P} , through



recursion as follows (we use $M(\Gamma)$ to denote the collection of maximally consistent subsets of Γ):

$$\begin{aligned}
 P_1 &= \begin{cases} \{\Gamma \cap \Sigma_1\} & \text{if } \Gamma \cap \Sigma_1 \text{ is consistent} \\ \{\Delta \mid \Delta \in M(\Gamma \cap \Sigma_1)\} & \text{else} \end{cases} \\
 &\quad \vdots \\
 P_i &= \begin{cases} \{(\Gamma \cap \Sigma_i) \cup \Pi \mid \Pi \in P_{i-1}\} & \text{if } (\Gamma \cap \Sigma_i) \cup \Pi \text{ is consistent} \\ \{\Pi \cup \Delta \mid \Pi \in P_{i-1} \text{ and} \\ \Pi \cup \Delta \in M((\Gamma \cap \Sigma_i) \cup \Pi)\} & \text{else} \end{cases}
 \end{aligned}$$

Let $P = P_n$

Informally, the set of preferred subsets of Γ is constructed by first picking out members of Γ that are in the first $n - 1$ strata of a modal family \mathcal{M} . This selected subset is clearly consistent since Σ_{n-1} is consistent. The selected subset is then consistently extended to maximality with the remaining members of Γ . Clearly we are not guaranteed to have a single preferred subset in every case. In one sort of cases, a report A may directly contradict our theory. But it is also possible that competing claims may each be consistent with our theory and so there may be more than one preferred subset.

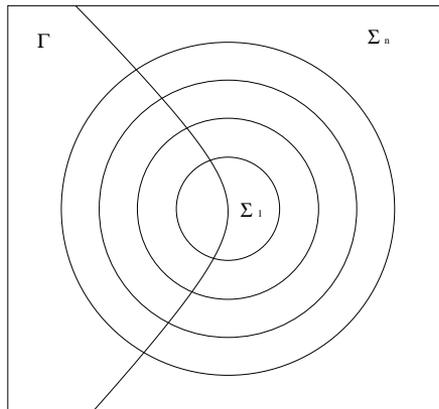


Figure 1. Rescher's Hypothetical Reasoning



In applying classical reasoning to our preferred subsets, there are clearly two choices: we may accept a claim as long as it is a classical conclusion of *some* preferred subset, or alternatively we may accept a claim as long as it is a classical conclusion of *every* preferred subset.

DEFINITION 2.3. Relative to an arbitrary but fixed modal family $\mathcal{M} = \langle \Sigma_1, \dots, \Sigma_n \rangle$,

- A is a U -consequence (universal-consequence) of Γ iff for every $\Pi \in \mathbf{P}$, $\Pi \vdash A$, and
- A is a E -consequence (existential-consequence) of Γ iff for some $\Pi \in \mathbf{P}$, $\Pi \vdash A$.

We should note that if \mathcal{M} is just $\langle \Phi \rangle$, then the U -consequences (E -consequences) of Γ are just formulae classically derivable from every (some) maximally consistent subset of Γ .¹ In the remaining, we'll make the simplifying assumption that $\mathcal{M} = \langle \Phi \rangle$.

We note that Rescher's formalism for U -consequence (resp. E -consequence) in fact corresponds to a special class of (propositional) default logics called *prerequisite-free normal* default theories. In [38, 15], Stillman and Gottlob independently show that the complexity of the decision problem(s) for deciding whether a formula is a *credulous* (*skeptical*) consequence of a *prerequisite-free normal* default theory is Σ_2^P -complete (Π_2^P -complete). Given these results, deciding U -consequence (E -consequence), in the worst case, is likely to require an NP -oracle exponential amount of time to solve.

3. Belnap's Conjunctive Containment

In [2], Belnap considered a strategy to improve the Rescher-mechanism by finding different *articulations* for a set of logical descriptions. Recall that Belnap's main criticism of Rescher is that reasoning with maximal consistent subsets is too syntax dependent in the sense that a minor syntactic variant of a logical representation may yield wildly different conclusions. Belnap's main idea is that given a set of data Γ we can *pre-process* Γ with certain closure operations so that the *content* of the data

¹ E -consequences (U -consequences) are typically called *credulous* (*skeptical*) consequences in the default logic literature.

can be made explicit and information not involved in any inconsistency, the ‘innocent part’ of the data, can be isolated. Once this is done, we can then apply the Rescher-mechanism to reason with the extended data set. Indeed Belnap’s suggestion is not fundamentally different from the methodology of *knowledge compilation* in AI (see [22, 23]). In knowledge compilation, the general aim is to give a sound and complete *translation* of data represented in a general language to a target language with lower computational complexity. The translation is done *off-line* so that the computational cost of inference is shifted from run time query-answering to off-line compilation. Thus in knowledge compilation, reasoning can be viewed as a two stage process involving both data preparation and formal deduction from prepared data. For Belnap however, the concern is not so much to reduce the computational cost but to reduce the effect of syntactic variations on inferences from the innocent part of the data set. Clearly this can be seen as a form of data preparation. In both cases, the key step is to perform some sort of *content preserving operations* on a data set prior to the application of (automated) deduction.

It is instructive to recap Belnap’s reasons for rejecting the use of the first degree entailment (*FDE*) of the relevant logic R , Parry’s analytic implication (*AI*) and Angell’s analytic containment (*AC*) as candidate closures or *articulations* of the input data:

Example 3.1. $\Gamma = \{p, \neg p, q\}$

In *FDE* and Parry’s *AI* we have, respectively

$$A \vdash_{FDE} A \vee B \quad (1)$$

$$A \wedge B \vdash_{AI} A \vee \neg B \quad (2)$$

Hence the closure of Γ under either *FDE* or *AI* yields $\neg p \vee \neg q$ which conspires together with p to prevent q from being derived as a *U*-consequence.

Example 3.2. $\Gamma = \{p, \neg p, q, r \vee \neg q\}$

In Angell’s *AC* we have

$$A \wedge (B \vee C) \vdash_{AC} A \wedge (B \vee C) \wedge (A \vee C) \quad (3)$$

Hence we get $\neg p \wedge (r \vee \neg q) \vdash_{AC} \neg p \vee \neg q$. Once again $\neg p \vee \neg q$ conspires together with p to prevent q from being derived as a *U*-consequence.



In each of these cases, the use of a certain version of disjunction introduction results in the introduction of additional inconsistencies. Indeed this is symptomatic of the kind of difficulties involved in the use of disjunction introduction in the presence of inconsistencies. Recall that in C. I. Lewis' original proof, disjunction introduction in its unrestricted form (1), when combined with the use of *disjunctive syllogism* (*DS*), $A \wedge (\neg A \vee B) \vdash B$, yields the *ex falso* $A \wedge \neg A \vdash B$. In the case of Parry's Analytic Implication, the combination of (2) and *DS* yields a kind of inference which propagates inconsistencies:

$$(A \wedge \neg A) \wedge B \vdash (B \wedge \neg B) \quad (4)$$

We note that although B only occurs positively on the left of \vdash in (4), the inconsistency of B ends up being the conclusion. We may think of this as perhaps a case of guilt by association — B is hanging out with the wrong crowd here. In the case of Angell's Analytic Containment, the combination of (3) and *DS* deliver a more subtle way to propagate inconsistencies:

$$(A \wedge \neg A) \wedge (C \vee B) \wedge (D \vee \neg B) \vdash (B \wedge \neg B) \quad (5)$$

Note that in (5) both positive and negative occurrences of B on the left of \vdash are required to produce $B \wedge \neg B$ on the right. We also note that neither B nor $\neg B$ can be deduced without the use of the inconsistent A and $\neg A$.²

In recent years, Hewitt ([16, 17]) has investigated a logic (called Direct Logic) in which (4) is rejected. Hewitt describes Direct Logic as a logic of “strong paraconsistency”. Accordingly to Hewitt, “...for the purposes of reasoning about large software systems, a stronger principle is needed ... it requires $P, \neg P, Q \not\vdash \neg Q$ because the inconsistency between P and $\neg P$ is not relevant to Q ” ([16], page 113). As noted by Hewitt ([16], page 118), the propositional fragment of Direct Logic appears to be closely related to the relevant logic *R*-Mingle (i.e. the logic *R* with the Mingle Axiom $\vdash A \rightarrow (A \rightarrow A)$ added, a.k.a *RM*). While we may agree that most “paraconsistent logics” are unsuitable for reasoning about large software systems—rejection of the *ex falso* alone appears too

²Besnard and Hunter ([7, 19, 20, 21]) have developed an interesting strategy to block the harmful interaction between (1) and *DS*. In their Quasi-Classical (QC) logic, no proof is allowed to use *DS* subsequent to the use of (1).

weak to quarantine inconsistencies, however, we do not know whether Direct Logic or *R*-Mingle can circumvent the more subtle form of *inconsistencies spread* in principles like (5) either.

Returning to Belnap’s original line of argument, Belnap’s own conclusion is that none of *FDE*, *AI* and *AC* can be used as a basis to articulate the content of the inconsistent data—each of (1), (2) and (3) would provide an alternative route to contaminate the ‘innocent part’ of the data. To overcome this problem, Belnap proposes the use of *conjunctive containment*. First we recall some of Belnap’s definitions. To simplify the matter, we’ll assume that our language is restricted to the truth functional connectives $\{\neg, \wedge, \vee\}$.

DEFINITION 3.1. A subformula *B* of a given formula *A* is said to be a zero subformula if it is within the scope of zero negations, an even subformula if it is within the scope of zero or an even number of negations, otherwise it is said to be odd.

DEFINITION 3.2. Belnap’s replacement rules are given as follows:

$$[*] \frac{\dots(B \wedge C)\dots}{\dots B \dots \quad \dots C \dots}$$

provided that $(B \wedge C)$ is an even subformula.

$$[\#] \frac{\dots(B \vee C)\dots}{\dots B \dots \quad \dots C \dots}$$

provided that $(B \vee C)$ is an odd subformula.

Clearly for any given *A* we can build a finite binary tree \mathcal{T} such that (1) the root of \mathcal{T} is just *A*, (2) each branching is an application of either $[\#]$ or $[*]$, (3) a node is either the root of \mathcal{T} or a formula obtained by $[\#]$ or $[*]$, and (4) the leaves or end points are formulae which contain no even subformulae of the form $(B \wedge C)$ and no odd subformulae of the form $(B \vee C)$.

For convenience we shall draw a tree with the root at the bottom and all branches extending upward, i.e. we apply the replacement rules as if they are upside down. Since the order in which we apply $[\#]$ and $[*]$ can be permuted, clearly such a tree is not unique for a given *A* in general. But there can be at most finitely many such trees for a given *A*. Thus we can associate with each *A* a finite set of trees $\{\mathcal{T}_1^A, \dots, \mathcal{T}_n^A\}$



where each \mathcal{T}_i^A is a finite binary tree built in the prescribed way. We'll call these the Belnap trees associated with A .

LEMMA 3.1. *Let $(B \wedge C)$ be a zero subformula in $D = \dots(B \wedge C)\dots$, then D is classically equivalent to $D' = ((\dots B \dots) \wedge (\dots C \dots))$.*

PROOF. Since $(B \wedge C)$ is a zero subformula, we can equivalently transform D into the following form:

$$D_0 = \overbrace{(\dots (B \wedge C) *_1 A_1) *_2 A_2) *_3 \dots}^{n+1} *_n A_n)$$

where $*_i$ is either \wedge or \vee .

Since both \wedge and \vee are commutative, we note any step in the transformation from D to D_0 is reversible and equivalence preserving. We'll denote the transformation from D to D_0 as T and the reverse of T as T' . We'll show by induction on the depth d of $(B \wedge C)$, defined in terms of the number of ' \wedge ' to the left of $B \wedge C$, that D_0 is equivalent to

$$D'_0 = [\overbrace{(\dots (B *_1 A_1) *_2 \dots) *_n A_n}^n \wedge \overbrace{(\dots (C *_1 A_1) *_2 \dots) *_n A_n}^n]$$

For the basis $d = 1$: this is trivial since $(B \wedge C)$ is equivalent to itself. For the inductive step, we'll make the assumption that the statement holds for $d = k$ and show that it holds for the case when $d = k + 1$. Since we assume that $d = k + 1$, D_0 must be of the form:

$$\overbrace{(\dots (B \wedge C) *_1 A_1) *_2 A_2) *_3 \dots}^{k+1} *_k A_k)$$

By the induction hypothesis, the following subformula of D_0

$$\overbrace{(\dots (B \wedge C) *_1 A_1) *_2 A_2) *_3 \dots}^k *_k A_k)$$

is equivalent to

$$[\overbrace{(\dots (B *_1 A_1) *_2 \dots) *_k A_k}^k \wedge \overbrace{(\dots (C *_1 A_1) *_2 \dots) *_k A_k}^k]$$

Hence D_0 must be equivalent to



$$E = ([\overbrace{(\dots (B *_1 A_1) *_2 \dots)}^k *_k A_k] \wedge \overbrace{(\dots (C *_1 A_1) *_2 \dots)}^{k-1} *_k A_k)$$

There are two cases to consider: either $*_k$ is \wedge or \vee . In the first case E is equivalent to

$$[\overbrace{(\dots (B *_1 A_1) \dots *_k A_k)}^{k+1} \wedge A_k] \wedge [\overbrace{(\dots (C *_1 A_1) \dots *_k A_k)}^{k+1} \wedge A_k]$$

In the later case, using distribution of \vee over \wedge , E is equivalent to

$$[\overbrace{(\dots (B *_1 A_1) \dots *_k A_k)}^{k+1} \vee A_k] \wedge [\overbrace{(\dots (C *_1 A_1) \dots *_k A_k)}^{k+1} \vee A_k]$$

This suffices to show that D_0 and D'_0 are equivalent. To complete the proof we make use of the fact that the transformation T from D to D_0 is reversible. Hence by applying the reverse transformation T' to the left and right conjuncts of D'_0 , the equivalence of D and D' follows. \dashv

PROPOSITION 3.1. *If E and F are obtained from A by an application of either $[*]$ or $[\#]$, then $E \wedge F$ is classically equivalent to A .*

PROOF. If E and F are obtained from A by an application of $[*]$, then E and F must be obtained via an even subformula $(B \wedge C)$ of A . Since $(B \wedge C)$ is even, repeat applications of pushing negations onto $(B \wedge C)$ will result in an even number of occurrences of negation in front of $(B \wedge C)$. By double negation elimination we can transform A into an equivalent formula A' where $(B \wedge C)$ is a zero subformula. Using Lemma 3.1 and reversing the relevant transformation steps, the desired result follows.

If E and F are obtained via $[\#]$, then E and F must be obtained via an odd subformula $(B \vee C)$ of A . Since $(B \vee C)$ is odd, repeat application of pushing negation onto $(B \vee C)$ will result in an odd occurrences of negation in front of $(B \vee C)$. Using double negation elimination repeatedly and pushing the remaining negation into $(B \vee C)$ will result in an even subformula $(\neg B \wedge \neg C)$. Again applying Lemma 3.1 and reversing all relevant transformation steps, the desired result follows. \dashv

COROLLARY 3.1. *For any Belnap tree \mathcal{T}^A associated with A , A is classically equivalent to the conjunction of all the leaves in \mathcal{T}^A .*



DEFINITION 3.3. Belnap's Closure, C_B , on a given A is defined as follows: $D \in C_B(A)$ iff D is a node of some Belnap tree associated with A . For a given set of formulae Γ , $C_B(\Gamma) = \{D \in C_B(A) : A \in \Gamma\}$.

A set Γ conjunctively contains A in the strict sense iff $A \in C_B(\Gamma)$, i.e. A is a node of some Belnap tree associated with some $B \in \Gamma$.

The extended Belnap's Closure, C_B^+ on a set Γ is defined as follows: $A \in C_B^+(\Gamma)$ iff every member of $C_B(A)$ is classically equivalent to a conjunction of some members of $C_B(\Gamma)$.

Alternatively we may define $C_B(\Gamma)$ simply as the least superset of Γ that is closed under $[*]$ and $[\#]$. However, the use of Belnap's tree gives us an easy way to visualise the underlying mechanism: each piece of information A is *conjunctively eliminated* at each level of a Belnap's tree until all hidden conjunctions are eliminated; since $[*]$ and $[\#]$ preserve the (classical) model(s) of their premises, all information implicitly encoded in A is successively passed on to the next level in its Belnap's tree.

With respect to the extended Belnap's closure C_B^+ , the basic idea is to regain some limited form of conjunction introduction with members of C_B while adding all those that are classically equivalent to these conjunctions without creating unexpected nonequivalence.

Example 3.3. Let $\Gamma = \{p\}$. We have $p \wedge p \in C_B^+(\Gamma)$ but $p \vee (p \wedge q) \notin C_B^+(\Gamma)$ even though $p \wedge p$ is classically equivalent to $p \vee (p \wedge q)$. Note that although we have $p \vee q \in C_B(\{p \vee (p \wedge q)\})$, $p \vee q$ is not classically equivalent to any conjunction of members of $C_B(\Gamma)$.

FACT 3.1. C_B and C_B^+ are closure operators in the sense of Tarski, i.e. they satisfy inclusion, monotonicity and idempotence. C_B^+ is an extension of C_B , i.e. for any $\Gamma \subseteq \Phi$, $C_B(\Gamma) \subseteq C_B^+(\Gamma)$. Moreover they distribute over union, i.e. $C_B(\Gamma \cup \Gamma') = C_B(\Gamma) \cup C_B(\Gamma')$

While we may think of Belnap's closure C_B as an articulation of a set of formulae, the extension C_B^+ is a *proper E-equivalent extension* of C_B in the following sense:

DEFINITION 3.4. A closure operator C is a proper E -equivalent extension of a closure operator C' iff for any premise set $\Gamma \subseteq \Phi$,

1. $C(\Gamma)$ and $C'(\Gamma)$ have exactly the same set of existential-consequences, i.e. $C_E^*(C(\Gamma)) = C_E^*(C'(\Gamma))$.

2. $C'(\Gamma) \subseteq C(\Gamma)$.
3. $C(C'(\Gamma)) = C(\Gamma)$.

If condition (1) holds for C , then we say that C is E -equivalent to C' .

LEMMA 3.2. *For an arbitrary but fixed Γ , let $M(C_B(\Gamma))$ and $M(C_B^+(\Gamma))$ be the collections of maximal consistent subsets of $C_B(\Gamma)$ and $C_B^+(\Gamma)$ respectively. Then there is a bijection f with domain $M(C_B(\Gamma))$ and range $M(C_B^+(\Gamma))$ such that for any $\mathcal{A} \in M(C_B(\Gamma))$, $f(\mathcal{A})$ is classically equivalent to \mathcal{A} .*

PROOF. Let Γ be arbitrary but fixed. Let

$$\begin{aligned} M(C_B(\Gamma)) &= \{\mathcal{A}_i : i \in I\} \\ M(C_B^+(\Gamma)) &= \{\mathcal{B}_j : j \in J\} \end{aligned}$$

We observe that:

1. For each $i \in I$ there exists a $j \in J$ such that $\mathcal{A}_i \subseteq \mathcal{B}_j$: by the consistency of \mathcal{A}_i and the fact that $C_B(\Gamma) \subseteq C_B^+(\Gamma)$.

2. For each $i \in I$ there is exactly one $j \in J$ such that $\mathcal{A}_i \subseteq \mathcal{B}_j$: from (1) the existence of such a \mathcal{B}_j is guaranteed for each arbitrary but fixed $i \in I$. Toward a contradiction assume that for some $k \in J$, $k \neq j$, $\mathcal{A}_i \subseteq \mathcal{B}_k$. Note that since $\mathcal{B}_j \neq \mathcal{B}_k$, $\mathcal{B}_j \cup \mathcal{B}_k$ is inconsistent. Hence there exists $D_1, \dots, D_m \in \mathcal{B}_j$ and $E_1, \dots, E_n \in \mathcal{B}_k$ such that

$$D_1 \wedge \dots \wedge D_m \vdash \neg(E_1 \wedge \dots \wedge E_n)$$

We claim that every member of $C_B(D_1) \cup \dots \cup C_B(D_m)$ must be classically equivalent to a conjunction of some members of \mathcal{A}_i . Suppose not. Then there must be a member of $C_B(D_1) \cup \dots \cup C_B(D_m)$ classically equivalent to a conjunction of members involving elements of $(C_B(\Gamma) \setminus \mathcal{A}_i)$. But this is impossible since by the maximal consistency of \mathcal{A}_i any $A \in C_B(\Gamma) \setminus \mathcal{A}_i$ is inconsistent with \mathcal{A}_i and this would imply that \mathcal{A}_i is inconsistent with \mathcal{B}_j . Similar argument also shows that every member of $C_B(E_1) \cup \dots \cup C_B(E_n)$ must be classically equivalent to a conjunction of some members of \mathcal{A}_i . But this clearly contradicts the consistency of \mathcal{A} . Hence \mathcal{B}_j cannot be distinct from \mathcal{B}_k afterall.

3. For no $i, i' \in I$, $i \neq i'$ do we have $\mathcal{A}_i \subseteq \mathcal{B}_j$ and $\mathcal{A}_{i'} \subseteq \mathcal{B}_j$ for some $j \in J$: by the consistency of each \mathcal{B}_j .

4. For each $j \in J$ there exists an $i \in I$ such that $\mathcal{A}_i \subseteq \mathcal{B}_j$: it is straightforward to verify that for each $j \in J$, $\mathcal{B}_j \cap C_B(\Gamma)$ is a maximal consistent subset of $C_B(\Gamma)$.

We now define the function $f: C_B(\Gamma) \rightarrow C_B^+(\Gamma)$ as follows: for each $i \in I$

$$f(\mathcal{A}_i) = \mathcal{B}_j \Leftrightarrow \mathcal{A}_i \subseteq \mathcal{B}_j$$

for some $j \in J$. Clearly by observation (2), f is a well defined function. By observation (4), f is surjective. By observation (3) f is injective. Hence f is a bijection.

Finally to show that for every $i \in I$, \mathcal{A}_i and $f(\mathcal{A}_i)$ are classically equivalent, it suffices to observe that the argument for observation (2) establishes that for every $A \in f(\mathcal{A}_i)$, every member of $C_B(A)$ is classically equivalent to a conjunction of some members of \mathcal{A}_i . \dashv

PROPOSITION 3.2. C_B^+ is a proper E and U -equivalent extension of C_B .

PROOF. By Lemma 3.2, condition (1) of Definition 3.4 is clearly satisfied. Moreover $C_B(\Gamma) \subseteq C_B^+(\Gamma)$ clearly holds. It remains to verify that for any Γ , $C_B^+(C_B(\Gamma)) = C_B^+(\Gamma)$.

(\supseteq): Since C_B and C_B^+ are both Tarskian closure operators, we have $\Gamma \subseteq C_B(\Gamma)$ and hence $C_B^+(\Gamma) \subseteq C_B^+(C_B(\Gamma))$.

(\subseteq): If $A \in C_B^+(C_B(\Gamma))$, then for every $B \in C_B(A)$, there are $C_1, \dots, C_n \in C_B(C_B(\Gamma))$ such that B is classically equivalent to $C_1 \wedge \dots \wedge C_n$. But $C_B(C_B(\Gamma)) = C_B(\Gamma)$, hence $A \in C_B^+(\Gamma)$ as required. \dashv

Lemma 3.2 clearly gives a sufficient condition for E -equivalence—two sets of formulae (with arbitrary cardinalities) are E -equivalent if a bijection of the suitable sort exists between the two collections of maximal consistent subsets of the two sets. But it is unclear that this is also necessary in cases where infinite cardinalities are considered. In [36], Rescher and Manor give the necessary and sufficient conditions for E -equivalence for *finitely generated* sets of formulae. But no general characterisation is given there. For sets with finitely many maximal consistent subsets (though not necessarily finitely generated in Rescher's sense), the following proposition gives a necessary condition for their E -equivalence:

PROPOSITION 3.3. Let $|M(\Gamma)| < \omega$. If Γ' is E -equivalent to Γ , then $|M(\Gamma')| = |M(\Gamma)|$. Equivalently, for any Γ and Γ' , if $|M(\Gamma')|$ and $|M(\Gamma)|$ are finite and $|M(\Gamma')| \neq |M(\Gamma)|$, then Γ and Γ' are not E -equivalent.

PROOF. Without loss of generality, we may assume that Γ and Γ' are such that $|M(\Gamma)| = n$ and $|M(\Gamma')| = m$ where $m < n$. Towards a contradiction, we assume that Γ and Γ' are E -equivalent. We let $M(\Gamma) = \{\mathcal{A}_1, \dots, \mathcal{A}_n\}$ and $M(\Gamma') = \{\mathcal{B}_1, \dots, \mathcal{B}_m\}$. Since members of $M(\Gamma')$ are pairwise inconsistent, by the standard compactness theorem, there exist formulae A_1, \dots, A_n such that for every $i \leq n$,

1. $\mathcal{A}_i \vdash A_i$ and
2. $A_i \vdash \bigwedge_{i \neq j} \neg A_j$

By our reductio assumption Γ and Γ' are E -equivalent and hence by (1) above for each $i \leq n$ there exists a $k \leq m$ such that $\mathcal{B}_k \vdash A_i$. However by our initial assumption $m < n$ and thus by the pigeonhole principle, there exists a $t \leq m$ such that $\mathcal{B}_t \vdash \bigwedge_{i \leq n} \neg A_i$. Clearly by the consistency of each $\mathcal{A}_i \in M(\Gamma)$, $\mathcal{A}_i \not\vdash \bigwedge_{i \leq n} \neg A_i$. But this contradicts the assumption that Γ and Γ' are E -equivalent. This suffices to show that Γ and Γ' are not E -equivalent on the assumption that $n \neq m$. \neg

Thus for any two finite sets, we can test for their non- E -equivalence by simply counting their number of maximal consistent subsets.

4. An Improvement to Belnap's Strategy

One of the main motivations for Belnap to introduce C_B is to provide a standard way to isolate the effect of the inconsistencies in a data set. In [18], Horty explicitly endorsed a similar strategy for handling inconsistent instructions using modalised versions of Belnap's replacement rules:

$$[\Box*] \frac{\Box(\dots(B \wedge C)\dots)}{\Box(\dots B \dots) \quad \Box(\dots C \dots)} \quad [\Box\#] \frac{\Box(\dots(B \vee C)\dots)}{\Box(\dots B \dots) \quad \Box(\dots C \dots)}$$

where ' \dots ' does not contain any occurrence of ' \Box ' and the even and odd restrictions again apply to the respective rule.

To appreciate the significance of the $[*]$ and $[\#]$ rules (and their modalised counterparts), we quote a remark of Belnap:

Since different ways of articulating our beliefs [...] give different results under Rescher's proposal and since we do not want this, evidently we have to have some views about which articulations we

most want to reflect [...] Policy: try to reflect *maximum* articulation. [...] if we maximally articulate [...] we may be able to isolate the effect of its contradiction, [...] [o]r, which seems just as important, we may be able to block a consequence by freeing for use some conjunct of a conjunction which is itself not consistently available [...]. [2, p. 545]

In light of the [*] and [#] rules, *maximum* articulation here is cashed out in terms of conjunction elimination. In certain cases this is just the right remedy. Consider for instance:

Example 4.1. Let $A = (p \wedge \neg p) \wedge (q \vee r)$ and let $\Gamma = \{A, \neg r, \neg p\}$.

Applying the [*] rule to A we get

$$\frac{\frac{p \quad \neg p}{p \wedge \neg p} \quad q \vee r}{(p \wedge \neg p) \wedge (q \vee r)}$$

Figure 2. Belnap tree for $(p \wedge \neg p) \wedge (q \vee r)$

In Example 4.1, all conjunctions are eliminated to maximally articulate the information encoded by A . The result is that the inconsistency with respect to p would neither interfere with $\neg r$ nor $q \vee r$. We note however that an imprudent use of [*] may result in duplication and thereby increase the size of the tree (see Figure 3).

$$\frac{\frac{p \quad q \vee r}{p \wedge (q \vee r)} \quad \frac{\neg p \quad q \vee r}{\neg p \wedge (q \vee r)}}{(p \wedge \neg p) \wedge (q \vee r)}$$

Figure 3. Belnap tree for $(p \wedge \neg p) \wedge (q \vee r)$

However, our main concern here is not with efficiency. Our main concern is that there are cases in which [*] and [#] cannot eliminate conjunctions without a *detour* in using the distributive properties of \wedge over \vee and vice versa. Consider for instance a slight variant of Example 4.1:

Example 4.2. Let $B = \neg[(p \vee \neg p) \vee \neg q] \vee \neg[(p \vee \neg p) \vee \neg r]$ and $\Gamma' = \{B, \neg r, \neg p\}$.



Assuming that we have the usual double negation elimination rule, contraction for \vee and \wedge , and commutative and associative rules for \vee and \wedge , we can now apply $[\#]$ to B in Example 4.2 to obtain the Belnap tree for B (Figure 4).

p	$\neg p$	$\neg p \vee r$	$p \vee r$	$q \vee r$	$p \vee q$	$\neg p \vee q$
$\neg(p \vee \neg p)$	$\neg[(p \vee \neg p) \vee \neg r]$	$\neg r$	$\neg r$	$\neg r$	$\neg r$	$\neg r$
$\neg(p \vee \neg p) \vee \neg[(p \vee \neg p) \vee \neg r]$	$\neg r$	$\neg r$	$\neg r$	$\neg r$	$\neg r$	$\neg r$
$\neg[(p \vee \neg p) \vee \neg q] \vee \neg[(p \vee \neg p) \vee \neg r]$						

Figure 4. Belnap tree for $\neg[(p \vee \neg p) \vee \neg q] \vee \neg[(p \vee \neg p) \vee \neg r]$

This is not entirely satisfactory because in taking an unnecessary detour, we have produced additional disjunctive information. Belnap’s initial objection against the use of relevant implication and analytic implication is precisely that closures under these implications are too liberal in generating disjunctive information. The point is that disjunctive information can interact with inconsistencies in such a way that further inconsistencies can be produced. In the presence of $\neg r$ for instance, $p \vee r$ and $\neg p$ form an inconsistent triad. Comparing this with Figure 3 however, $\neg r$ remains innocent. We note further that the *distributivity* of \wedge over \vee and \vee over \wedge are built into Belnap’s replacement rules—we cannot avoid the use of distributivity with these rules.

While our previous example demonstrates how distributivity is used in the context of implicit conjunction, our next example shows that the same is true of explicit conjunction:

Example 4.3. Let $C = (p \wedge q) \vee (p \wedge r)$ and $\Gamma = \{C, \neg r, \neg p\}$.

Applying the $[\ast]$ rule to C we get the following Belnap tree:

p	$p \vee r$	$p \vee q$	$q \vee r$
$p \vee (p \wedge r)$	$q \vee (p \wedge r)$		
$(p \wedge q) \vee (p \wedge r)$			

Figure 5. Belnap tree for $(p \wedge q) \vee (p \wedge r)$



$$\frac{\frac{p \quad q \vee r}{p \wedge (q \vee r)}}{(p \wedge q) \vee (p \wedge r)}$$

Figure 6. Factoring for $(p \wedge q) \vee (p \wedge r)$

The example here (Figure 5) is similar to the previous case: $p \vee r$ conspires with $\neg p$ in Γ to prevent $\neg r$ from being derived as an U -consequence. Contrasting this with the case where we use a more direct route to eliminate conjunction while keeping disjunctive information to a minimal (Figure 6), we get a very different result. But what can we say coherently about these examples? There are (at least) two possible options:

1. Our examples do encode different information in each instance and hence conjunctive containment merely makes explicit the difference. This is reflected in the production of different disjunctive information under conjunctive containment.
2. Our examples do not encode substantively different information in each instance. The fact that their conjunctive closures differ shows that conjunctive containment still over generates—in particular it over generates by producing too much disjunctive information.

Our intuition in this matter may not run very deep. Indeed there may not be any definitive reason to settle for one over the other. While option (1) is certainly a coherent position (and we suspect this is the option Belnap is likely to take), we would like to explore option (2) here and flesh out an account where examples 4.1, 4.2, and 4.3 do not yield different U -consequences while their maximal articulations are narrower than conjunctive containment.

4.1. Logic Minimisation

We start with Example 4.3 first. We note that the set of leaves $\Delta' = \{p, q \vee r\}$ in Figure 6 is consistent and classically equivalent to the set of leaves $\Delta = \{p, p \vee r, p \vee q, q \vee r\}$ in Figure 5. However, they clearly differ in the way in which they interact with the remaining members

of Γ . Δ would generate more inconsistent subsets when added to Γ than would Δ' . Furthermore, we note that every member of Δ' is a *prime implicate* of C . We'll briefly recap some of the standard definitions:

DEFINITION 4.1. A literal is either a propositional atom or the negation of a propositional atom. A disjunction of literals is said to be a clause. A clause D is an implicate of A iff $A \models D$. An implicate D of A is prime iff for all implicates D' of A if $D' \models D$ then $D \models D'$. A set of prime implicates $\{D_1, \dots, D_n\}$ of A is complete iff $\{D_1, \dots, D_n\} \models A$. A set of prime implicates $\{D_1, \dots, D_n\}$ is independent iff for no D_i do we have $\{D_1 \dots, D_{i-1}, D_{i+1} \dots D_n\} \models D_i$

The notion of prime implicate was (indirectly) introduced by Quine in [27, 28, 29].³ Quine was interested in simplifying truth functions and he showed that notion of prime implicate plays a central role in simplifying truth functions and thereby contributed directly to the minimisation and design of digital circuits. The emphasis on minimisation stems from the days when the production of logic gates was expensive and required considerable physical space and power. With the advent of semiconductor processes and VLSI, it is of course no longer a central concern to reduce the actual gate count for a system. Circuit design today is more concerned with physical space allocation, reliability and the correctness of a system. Interest in the use of prime implicates in circuit design has decreased considerably as a result. But the notion of prime implicate enjoys a renewed interest in recent years in light of works by de Kleer *et al* in logic based diagnostic systems [9, 10, 11].

Returning to our example however, it is easy to see that Δ' is a complete and independent set of prime implicates of C . It is thus natural to take Δ' to be the maximal articulation of C in Example 4.3. Our choice can be justified on the grounds that

- Δ' is a more compact representation of C ,
- Δ' minimises redundancies and disjunctive information, and
- Δ' minimises interference with $\neg r$.

³The author would like to thank Lloyd Humberstone for pointing out the confusion between the prime *implicate* and prime *implicant* in WCP4 presentation. Quine actually introduced the dual notion of prime implicant in [27, 28, 29].

However, the standard notion of prime implicate would not be able to handle Example 4.2 since Definition 4.1 uses a classical notion of consequence and thus inconsistent formulae would have the same (complete equivalence class) of prime implicates—namely the empty clause \emptyset .⁴ But this is exactly what we are trying to avoid in the first place. However, there is a straightforward way to amend the situation—we use a *relevant* notion of prime implicate:

DEFINITION 4.2. A clause D is a relevant implicate of A iff $A \models_{FDE} D$. A relevant implicate D of A is prime iff for all relevant implicates D' of A if $D' \models_{FDE} D$ then $D \models_{FDE} D'$. A set of relevant prime implicates $\{D_1, \dots, D_n\}$ of A is complete iff $D_1 \wedge \dots \wedge D_n \models_{FDE} A$. A set of relevant prime implicates $\{D_1, \dots, D_n\}$ is independent iff for no D_i do we have $D_1 \wedge \dots \wedge D_{i-1} \wedge D_{i+1} \wedge \dots \wedge D_n \models_{FDE} D_i$. We say that two sets of formulae Γ and Δ are *FDE* equivalent, written as $\Gamma \equiv_{FDE} \Delta$, iff $\bigwedge \Gamma \models_{FDE} \bigwedge \Delta$ and $\bigwedge \Delta \models_{FDE} \bigwedge \Gamma$.

The dual notion of *relevant prime implicant* of a given formula A can easily be defined: a relevant implicant of a formula A is a *cube* C (conjunction of literals) which *FDE*-entails A . In addition, C is *prime* if it is a minimal cube that *FDE*-entails A .

Since *FDE* is a paraconsistent logic, the empty clause is not a *FDE*-consequence of any (non-empty) inconsistent formula, i.e. for no A do we have $A \models_{FDE} \emptyset$. Given that *resolution* is not a valid form of inference in relevant logic in general, it is easy to see that the set of classical prime implicates (*PI*) and the set of relevant prime implicates (*RPI*) may be distinct for a given formula:

Example 4.4. $A = (q \vee r) \wedge ((p \vee q) \wedge (\neg p \vee q))$

⁴There have been other attempts in defining the *content* of propositional data. In [13, 14], Gemes gives several definitions of the *empirical content* of propositional data. However, none of his definition addressed the problem of inconsistent data and thus provided no useful means to individuate the content of inconsistent data. In [39], Tennant applies the Fregean Premise Principle for Content Identity to a relevant logic and shows that even with a relevant logic all contradictions are, in a sense, content equivalent. The author is inclined to take Tennant's result to be a *modus tollens* for rejecting either the Fregean Principle or the use of Tennant's relevant logic for specifying contents. In a separate paper, we'll investigate the possibility of saving the Fregean Principle using the proposal stated in this paper while maintaining that contradictions are NOT content equivalent.



	<i>RPI</i>	<i>PI</i>
$q \vee r$	✓	×
q	×	✓

Figure 7. *RPI* and *PI* of A

Although not every *RPI* of a given A is a *PI* of A , it is easy to see that every *RPI* of a given A is a classical implicate of A (since $\models_{FDE} \subset \models$).

Since any two complete independent sets of relevant prime implicates of a given formula must be *FDE* equivalent, we can treat them as unique up to equivalence. We'll use the notation $RPI(A)$ to denote any such complete independent set of relevant prime implicates of A . Similarly we use $PI(A)$ for the complete independent set of classical prime implicates of A . We note that $RPI(A)$ is a minimal set (ordered under \subseteq) that is both complete and independent. In classical logic, two formulae are equivalent iff their prime implicates are equivalent. This is also true with respect to *FDE* formulae:

PROPOSITION 4.1. For any A and B , $A \models_{FDE} B$ and $B \models_{FDE} A$ iff $RPI(A) \equiv_{FDE} RPI(B)$.

PROOF. (\Rightarrow): Suppose A and B are *FDE* equivalent. Let $RPI(A) = \{D_1, \dots, D_m\}$ and $RPI(B) = \{E_1, \dots, E_n\}$. By the transitivity of \models_{FDE} we have, $\bigwedge_{i \leq m} D_i \models_{FDE} E_j$ for each $j \leq n$. Hence $\bigwedge_{i \leq m} D_i \models_{FDE} \bigwedge_{j \leq n} E_j$. Similarly we can show that $\bigwedge_{j \leq n} E_j \models_{FDE} \bigwedge_{i \leq m} D_i$.

(\Leftarrow): Suppose $RPI(A) \equiv_{FDE} RPI(B)$. Then we have

$$A \models_{FDE} \bigwedge_{i \leq m} D_i \models_{FDE} \bigwedge_{j \leq n} E_j \models_{FDE} B$$

Similarly we have

$$B \models_{FDE} \bigwedge_{j \leq n} E_j \models_{FDE} \bigwedge_{i \leq m} D_i \models_{FDE} A$$

By the transitivity of these entailments, it follows that A and B are *FDE* equivalent. \dashv

An immediate corollary is that standard reduction rules for *CNF* (*DNF*) conversion are *RPI* preserving:



COROLLARY 4.1. *The following equivalences hold:*

1. $RPI(\neg\neg A) \equiv_{FDE} RPI(A)$
2. $RPI(\neg(A \vee B)) \equiv_{FDE} RPI(\neg A \wedge \neg B)$
3. $RPI(\neg(A \wedge B)) \equiv_{FDE} RPI(\neg A \vee \neg B)$
4. $RPI(A \vee (B \wedge C)) \equiv_{FDE} RPI((A \vee B) \wedge (A \vee C))$

The minimality of an RPI ensures that a certain *transitivity* property of RPI holds:

PROPOSITION 4.2. *For any formulae A , B and C , if $C \in RPI(B)$ and $B \in RPI(A)$, then $C \in RPI(A)$.*

PROOF. Given that $B \in RPI(A)$, B must be a clause and thus $B \in RPI(B)$ holds trivially. So if $C \in RPI(B)$, $B \equiv_{FDE} C$ follows immediately from Definition 4.2. Hence $C \in RPI(A)$. \dashv

Just as Belnap's replacement rules can be used as a basis for defining the closure operators C_B and C_B^+ , RPI s too can be used as a basis for defining certain Tarskian closure operators:

DEFINITION 4.3. For any A and Γ , define

$$\begin{aligned} C_{RPI}(A) &= RPI(A) \cup \{A\} \\ C_{RPI}(\Gamma) &= \{B \in C_{RPI}(A) \mid A \in \Gamma\} \\ C_{RPI}^+(\Gamma) &= \bigcup_{\Delta \subseteq_{\text{fin}} C_{RPI}(\Gamma)} \{B \mid B \equiv_{FDE} \bigwedge \Delta\} \end{aligned}$$

PROPOSITION 4.3. C_{RPI} and C_{RPI}^+ are both Tarskian closure operators. Moreover, C_{RPI}^+ is an E -equivalent (U -equivalent) extension of C_{RPI} .

PROOF. Reflexivity: trivial since $A \in C_{RPI}(A)$ for every $A \in \Gamma$.

Monotonicity: Assume $\Gamma \subseteq \Delta$, then if $B \in C_{RPI}(\Gamma)$, there must exist some $A \in \Gamma$ such that $B \in C_{RPI}(A)$. But $A \in \Delta$ holds, so $B \in C_{RPI}(\Delta)$ as required.

Idempotence: $C_{RPI}(\Gamma) \subseteq C_{RPI}(C_{RPI}(\Gamma))$ is implied by the monotonicity of C_{RPI} above. For $C_{RPI}(C_{RPI}(\Gamma)) \subseteq C_{RPI}(\Gamma)$, we note that Proposition 4.2 gives us the transitivity property of RPI :

$$D \in C_{RPI}(C_{RPI}(\Gamma)) \implies \exists A \in C_{RPI}(\Gamma) : D \in C_{RPI}(A)$$



$$\begin{aligned}
&\implies \exists B \in \Gamma : A \in C_{RPI}(B) \\
&\implies D \in C_{RPI}(B) \\
&\implies D \in C_{RPI}(\Gamma)
\end{aligned}$$

Reflexivity and monotonicity for C_{RPI}^+ are straightforward. For idempotence, we verify that $C_{RPI}^+(C_{RPI}^+(\Gamma)) \subseteq C_{RPI}^+(\Gamma)$:

$$\begin{aligned}
A \in C_{RPI}^+(C_{RPI}^+(\Gamma)) &\implies \exists C_1, \dots, C_i \in C_{RPI}(C_{RPI}^+(\Gamma)) : \\
&A \equiv_{FDE} C_1 \wedge \dots \wedge C_i \\
&\implies \exists D_1, \dots, D_i \in C_{RPI}^+(\Gamma) : \\
&\forall j \leq i, C_j \in C_{RPI}(D_j) \\
&\implies \forall j \leq i, \exists E_j^1, \dots, E_j^m \in C_{RPI}(\Gamma) : \\
&D_j \equiv_{FDE} E_j^1 \wedge \dots \wedge E_j^m \\
&\implies \forall j \leq i, \exists F_j^1, \dots, F_j^m \in \Gamma : \\
&E_j^1 \in C_{RPI}(F_j^1), \dots, E_j^m \in C_{RPI}(F_j^m) \\
&\implies \forall j \leq i, C_j \in C_{RPI}(E_j^1 \wedge \dots \wedge E_j^m) \\
&\implies \forall j \leq i, \exists k : C_j \equiv_{FDE} E_j^k \\
&\implies \forall j \leq i, \exists F_j^k \in \Gamma : C_j \in C_{RPI}(F_j^k) \\
&\implies C_1, \dots, C_i \in C_{RPI}(\Gamma) \\
&\implies A \in C_{RPI}^+(\Gamma)
\end{aligned}$$

To show that C_{RPI}^+ is an E -equivalent extension of C_{RPI} , we need to show that

1. C_{RPI} and C_{RPI}^+ have the same E -consequences.
2. For any Γ , $C_{RPI}(\Gamma) \subseteq C_{RPI}^+(\Gamma)$.
3. For any Γ , $C_{RPI}^+(C_{RPI}(\Gamma)) = C_{RPI}^+(\Gamma)$.

(2) is trivial. For (1) we note that any FDE equivalent formula are also classically equivalent, so an argument similar to Lemma 3.2 suffices to show that C_{RPI}^+ is an E -equivalent (U -equivalent) extension of C_{RPI} . Finally we verify that $C_{RPI}^+(C_{RPI}(\Gamma)) = C_{RPI}^+(\Gamma)$:

(\supseteq): Trivial since C_{RPI} and C_{RPI}^+ are both Tarskian closure operators.

(\subseteq): We note that C_{RPI} is idempotent.

$$A \in C_{RPI}^+(C_{RPI}(\Gamma)) \implies \exists B_1, \dots, B_i \in C_{RPI}(C_{RPI}(\Gamma)) :$$

$$\begin{aligned}
 & A \equiv_{FDE} (B_1 \wedge \dots \wedge B_i) \\
 \implies & \exists B_1, \dots, B_i \in C_{RPI}(\Gamma) : \\
 & A \equiv_{FDE} (B_1 \wedge \dots \wedge B_i) \\
 \implies & A \in C_{RPI}^+(\Gamma) \quad \dashv
 \end{aligned}$$

We note that Definition 4.3 makes use of $RPI(A)$ for each A in a given set Γ , but $\bigcup_{A \in \Gamma} RPI(A)$ need not be an independent set of RPI s. In particular redundant (i.e. disjunctively implied) information can be spread across members of Γ . This motivates the following alternative definition:

DEFINITION 4.4. For any Γ and any clause C , we define $C \in RPI^*(\Gamma)$ iff

1. for some $A \in \Gamma$, $C \in RPI(A)$ and
2. for any $B \in \Gamma$ and clause D , if $D \in RPI(B)$ and $D \models_{FDE} C$, then $C \models_{FDE} D$

For any Γ ,

$$C_{RPI}^*(\Gamma) = RPI^*(\Gamma) \cup \Gamma$$

Membership for C_{RPI}^* is clearly more stringent than C_{RPI} —a clause C is in $RPI^*(\Gamma)$ only if C is a relevant prime implicate of some member of Γ and no other member of Γ entails a relevant prime implicate stronger than C . This definition is similar to Definition 4.2 for the RPI 's of an individual formula. However, C_{RPI}^* is not a closure operator in Tarski's sense. Although both reflexivity and idempotence remain intact, C_{RPI}^* does not have the usual monotonicity property.

Example 4.5. $\Gamma = \{p \wedge (q \vee r)\}$, $p \in C_{RPI}^*(\Gamma)$ and $q \vee r \in C_{RPI}^*(\Gamma)$. But $q \vee r \notin C_{RPI}^*(\Gamma')$ where $\Gamma' = \{p \wedge (q \vee r), q\}$.

The failure of monotonicity should not be regarded as a defect of C_{RPI}^* . Arguably, *implicit information* need not always increase monotonically with respect to supersets; C_{RPI}^* is a possible candidate for specifying the content of a given set of logical expressions. To illustrate the difference between C_{RPI}^* and C_{RPI} consider the following example:

Example 4.6. $\Gamma = \{p, (r \wedge \neg r) \wedge (p \vee q), \neg p\}$

Since $p \in C_{RPI}^*(\Gamma)$ we have $p \vee q \notin C_{RPI}^*(\Gamma)$. However $p \vee q \in C_{RPI}(\Gamma)$ given that $p \vee q \in RPI((r \wedge \neg r) \wedge (p \vee q))$. Note that in Example 4.6 q is

an E -consequence of $C_{RPI}(\Gamma)$ but not an E -consequence of $C_{RPI}^*(\Gamma)$. In general, C_{RPI}^* does not yield the same E -consequence (U -consequence) as C_{RPI} .

PROPOSITION 4.4. *For any Γ ,*

1. $C_{RPI}^*(\Gamma) \subseteq C_{RPI}(\Gamma)$
2. $C_{RPI}(C_{RPI}^*(\Gamma)) = C_{RPI}(\Gamma)$
3. $C_{RPI}^*(C_{RPI}(\Gamma)) = C_{RPI}(\Gamma)$

PROOF. For (1) it suffices to observe that $RPI^*(\Gamma) \subseteq \bigcup_{A \in \Gamma} RPI(A)$.

(2 \supseteq): Since $\Gamma \subseteq C_{RPI}^*(\Gamma)$, we have $C_{RPI}(\Gamma) \subseteq C_{RPI}(C_{RPI}^*(\Gamma))$ by the monotonicity of C_{RPI} .

(2 \subseteq): From (1) we have $C_{RPI}^*(\Gamma) \subseteq C_{RPI}(\Gamma)$ so by the monotonicity of C_{RPI} it follows that $C_{RPI}(C_{RPI}^*(\Gamma)) \subseteq C_{RPI}(C_{RPI}(\Gamma))$. By the idempotence of C_{RPI} we have $C_{RPI}(C_{RPI}^*(\Gamma)) \subseteq C_{RPI}(\Gamma)$.

(3): One direction follows from Definition 4.4 immediately. For the other direction consider an arbitrary but fixed Γ_0 . From (1) above we have $C_{RPI}^*(C_{RPI}(\Gamma_0)) \subseteq C_{RPI}(C_{RPI}(\Gamma_0))$. But from Proposition 4.3 C_{RPI} is a Tarskian closure operator and thus $C_{RPI}(C_{RPI}(\Gamma_0)) = C_{RPI}(\Gamma_0)$. Hence $C_{RPI}^*(C_{RPI}(\Gamma_0)) \subseteq C_{RPI}(\Gamma_0)$. Since Γ_0 was arbitrary we conclude that for any Γ , $C_{RPI}^*(C_{RPI}(\Gamma)) \subseteq C_{RPI}(\Gamma)$. \dashv

Returning to examples 4.1 and 4.2, Belnap's replacement rules are complete with respect to the given A and B in these examples, i.e. $C_{RPI}(A) \subset C_B(A)$ and $C_{RPI}(B) \subset C_B(B)$, but the generated implicates are not all prime. So Belnap's replacement rules are unsound with respect to relevant prime implicates. In the general case, Belnap's replacement rules are not complete since they are insufficient to transform formulae into clausal form. Clearly for clause reduction we need the additional rule, $\vdash \neg(B \wedge C) \leftrightarrow (\neg B \vee \neg C)$, to distribute negation over conjunction. However C_B^+ is complete with respect to RPI 's, i.e. for any Γ , we have $C_{RPI}(\Gamma) \subseteq C_B^+(\Gamma)$. We summarise the relationships of these closure operators in Figure 8.

To illustrate consider $\Gamma = \{(p \wedge q) \vee (p \wedge r), \neg(p \wedge q) \wedge s\}$. Clearly, $p \vee r \in C_B(\Gamma)$ but $p \vee r$ is not an RPI , so $p \vee r \notin C_{RPI}^+(\Gamma)$. Region (1) is non-empty. Moreover $\neg(p \wedge q) \in C_B(\Gamma)$ but $\neg p \vee \neg q \in C_{RPI}^+(\Gamma)$, so region (2) is non-empty. Example 4.6 shows that region (3) is non-empty and

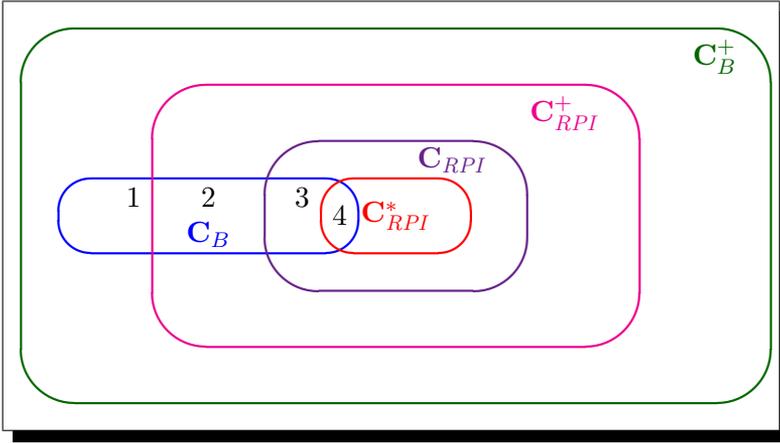


Figure 8. Relationships between Closure Operators

with minor modification it can show that region (4) is also non-empty. To see that $C_{RPI}^+ \subseteq C_B^+$, it suffices to note that

PROPOSITION 4.5. *For any clause D and formula A , if $D \in RPI(A)$, then $E \equiv_{FDE} D$ for some $E \in C_B(A)$.*

PROOF. We note that using arguments similar to proofs of Lemma 3.1 and Proposition 3.1, we can show that any A is FDE -equivalent to the conjunction of the leaves of the Belnap’s tree \mathcal{T}^A , i.e. $le(\mathcal{T}^A) \equiv_{FDE} RPI(A)$. Moreover, each leaf of a Belnap’s tree is conjunction free in the sense that each leaf is FDE -equivalent to a clause. Hence if $D \in RPI(A)$, there must be a leaf E of \mathcal{T}^A such that $D \equiv_{FDE} E$. \dashv

We should point out that in adopting the use of either C_{RPI} or C_{RPI}^+ for capturing the informational content of a formula, there is no guarantee that conjunction elimination is a sound strategy for generating RPI s. In general $RPI(A) \cup RPI(B) \neq RPI(A \wedge B)$.

Example 4.7. $A = p \wedge (p \vee q)$

In Example 4.7, it is clear that $RPI(p \wedge (p \vee q)) \subset RPI(p) \cup RPI(p \vee q)$. The containment here is proper. However, we do have the containment $RPI(A \wedge B) \subseteq RPI(A) \cup RPI(B)$ in the general case.

LEMMA 4.1. *For any clause C and any formula A and B , if $A \models_{FDE} C$, then $A \wedge B \models_{FDE} C$.*



PROOF. We use the ambi-valuation of Dunn [12] to prove our claim. Assume that $A \models_{FDE} C$. Then we have the implication $1 \in v(A) \Rightarrow 1 \in v(C)$ for any standard 4-valued valuation v of FDE . Consider an arbitrary v' such that $1 \in v'(A \wedge B)$. Then it follows that $1 \in v'(A)$ and $1 \in v'(B)$. So on v' in particular, $1 \in v'(C)$. Since v' was arbitrary, we have $A \wedge B \models_{FDE} C$ as required. \dashv

PROPOSITION 4.6. For any A and B , $RPI(A \wedge B) \subseteq RPI(A) \cup RPI(B)$.

PROOF. Assume that for an arbitrary clause D we have $D \in RPI(A \wedge B)$ but $D \notin RPI(A) \cup RPI(B)$. Then we have $A \wedge B \models_{FDE} D$ but $D \notin RPI(A)$ and $D \notin RPI(B)$. Then there are 4 cases to consider:

(Case 1) $A \not\models_{FDE} D$ and $B \models_{FDE} D$ but D is not prime for B : it follows that there exists a $D_0 \in RPI(B)$ such that $B \models_{FDE} D_0$ and $D_0 \models_{FDE} D$ but $D \not\models_{FDE} D_0$. By Lemma 4.1, we have $A \wedge B \models_{FDE} D_0$. But given that $D \in RPI(A \wedge B)$ and $D_0 \models_{FDE} D$, $D \models_{FDE} D_0$ holds. This is a contradiction.

(Case 2) $B \not\models_{FDE} D$ and $A \models_{FDE} D$ but D is not prime for A : the proof is similar to case 1 with B replaced with A throughout.

(Case 3) Both $A \models_{FDE} D$ and $B \models_{FDE} D$, but D is prime for neither A nor B : the argument in case (1) suffices to show that case 3 is impossible.

(Case 4) $A \not\models_{FDE} D$ and $B \not\models_{FDE} D$: we make use of the equivalence between FDE and *tautological entailment* as described in Anderson and Belnap ([1]). Since $A \not\models_{FDE} D$ and $B \not\models_{FDE} D$, for any arbitrary but fixed $DNF = C_1^A \vee \dots \vee C_m^A$ of A and $DNF = C_1^B \vee \dots \vee C_n^B$ of B , there exist some $i \leq m$, and some $j \leq n$ such that $C_i^A \not\models_{FDE} D$ and $C_j^B \not\models_{FDE} D$. Denote the set of literals occurring in C_i^A as $lit(C_i^A)$. We have $lit(C_i^A) \cap lit(D) = \emptyset$ and $lit(C_j^B) \cap lit(D) = \emptyset$. Hence $(lit(C_i^A) \cup lit(C_j^B)) \cap lit(D) = \emptyset$. Now consider the formula

$$E = \bigvee_{1 \leq k \leq m, 1 \leq l \leq n} (C_k^A \wedge C_l^B)$$

Clearly, E is a DNF of $A \wedge B$. Since $lit(C_i^A \wedge C_j^B) = (lit(C_i^A) \cup lit(C_j^B))$, we note that $lit(C_i^A \wedge C_j^B) \cap lit(D) = \emptyset$. We define a 4-valued assignment



v on the set of propositional atoms as follows:

$$\left\{ \begin{array}{ll} 0 \in v(p) \text{ and } 1 \notin v(p) & \text{if } \neg p \in \text{lit}(C_i^A \wedge C_j^B) \text{ and} \\ & p \notin \text{lit}(C_i^A \wedge C_j^B) \\ 1 \in v(p) \text{ and } 0 \notin v(p) & \text{if } p \in \text{lit}(C_i^A \wedge C_j^B) \text{ and} \\ & \neg p \notin \text{lit}(C_i^A \wedge C_j^B) \\ 1 \in v(p) \text{ and } 0 \in v(p) & \text{if } p \in \text{lit}(C_i^A \wedge C_j^B) \text{ and} \\ & \neg p \in \text{lit}(C_i^A \wedge C_j^B) \\ 1 \notin v(p) \text{ and } 0 \notin v(p) & \text{otherwise} \end{array} \right.$$

Clearly $1 \in v(C_i^A \wedge C_j^B)$ and hence $1 \in v(E)$ but by the disjointness of $\text{lit}(C_i^A \wedge C_j^B)$ and $\text{lit}(D)$, $1 \notin v(D)$. Hence $A \wedge B \not\models_{FDE} D$. But this contradicts the initial assumption that $D \in RPI(A \wedge B)$. \dashv

4.2. Algorithmic Considerations

Proposition 4.6 shows that in terms of using replacement rules in the style of $[*]$ or $[\#]$ for eliminating conjunctions, the RPI s of a child node need not be the RPI s of the root node. So although Corollary 4.1 shows that the standard reduction method for CNF conversion is indeed complete for generating RPI s, there is no guarantee that the clauses obtained are indeed independent. Checking for clause subsumption seems unavoidable and indeed critical when redundant information is presented. However when combined with a clause subsumption check, the standard CNF conversion algorithm can provide a sound and complete algorithm for generating RPI s.

Algorithm 4.1 RPI Generation

Require: input $A \in \Phi$

Ensure: output $S = RPI(A)$

- 1: convert A into $CNF(A)$ using the standard reduction method
 - 2: for each $C \in CNF(A)$, $S := S \cup \{C\}$ if C is relevant prime, else $S := S$.
 - 3: return S
-



Algorithm 4.1 is a naive method for generating *RPIs*. It first generates a set of relevant implicates of A and then prunes the set by removing all non-prime implicates. Clearly we have $CNF(A) \equiv_{FDE} RPI(A)$ given Corollary 4.1. So completeness is ensured in step (1) provided that step (2) does not remove implicates that are also prime (and clearly it doesn't). Although the clause subsumption check may be deployed earlier while $CNF(A)$ is generated, in the worst case the size of $CNF(A)$ can be exponentially related to the size of A , e.g. if $A = (p_1 \wedge p_2) \vee \dots \vee (p_{2n-1} \wedge p_{2n})$, there are 2^n clauses in the corresponding CNF . Our problem is inherently difficult computationally.

4.2.1. *PRI* via Classical *PI* Generation

In what follows, we'll present an alternative algorithm for generating $RPI(A)$ based on ideas from Ramesh *et al* [32, 30, 31] and Arieli and Denecker [3, 4]. The main idea here is to avoid the expensive CNF conversion by using *negated normal form* (NNF) instead. Once a formula A is converted into $NNF(A)$, we'll make use of Arieli and Denecker's *splitting transform* to convert $NNF(A)$ into a positive (i.e. negation free) formula $\widehat{NNF}(A)$.⁵ The conversion will preserve our problem in the sense that for any clause D , $D \in RPI(A)$ iff $\widehat{D} \in PI(\widehat{NNF}(A))$. So in effect our problem is transformed into the classical problem of prime implicate generation for a positive NNF formula. The algorithm of Ramesh *et al* [30, 31] can thus be invoked to generate the required *PI*'s via the use of the corresponding *semantic graph*.⁶ We recall some of the main definitions from Arieli and Denecker [3, 4].

DEFINITION 4.5. 1. A formula A is in negated normal form (NNF) iff no complex subformula of A is in the scope of a negation, i.e. only atomic formulae are within the scope of a negation operator.

2. Let $NNF(A)$ denotes the negated normal form of A . Then the splitting transform of $NNF(A)$, denoted by $\widehat{NNF}(A)$, is the formula obtained by uniformly substituting every unnegated atom p_i occurring

⁵We note that Besnard and Schaub [8] employed the same transform for defining signed systems of paraconsistent reasoning.

⁶We do not have space here to explicate the algorithm or the underlying concept of a *semantic graph* here. The reader can refer to Ramesh *et al* [30, 31] for full details.



in $NNF(A)$ with a new (signed) atom p_i^+ and every negated atom $\neg p_i$ in $NNF(A)$ with a new (signed) atom p_i^- [3, 4]. If $B = \widehat{A}$ for some A , then we define the inverse of splitting transform \overline{B} as the formula obtained by uniformly substituting every signed atom p_i^+ with literal p_i and every signed atom p_i^- with literal $\neg p_i$, i.e. $\overline{\widehat{A}} = A$

3. Let v be an arbitrary 4-valued assignment and $NNF(A)$ an arbitrary NNF formula.⁷ Then \widehat{v} is the 2-valued (classical) assignment defined as follows:

- For all p_i^+ and p_i^- occurring in $\widehat{NNF(A)}$, $\widehat{v}(p_i^+) = 1$ iff $1 \in v(p_i)$ and $\widehat{v}(p_i^-) = 1$ iff $0 \in v(p_i)$.

We note that both the splitting transform and \widehat{v} are well defined and do not depend on A . The following are consequences of Definition 4.5:

PROPOSITION 4.7.

1. Let v be an arbitrary 4-valued assignment and $NNF(A)$ be an arbitrary NNF formula. Let \widehat{v} be a 2-valued assignment as defined in (3) of Definition 4.5. Then $1 \in v(\widehat{NNF(A)})$ iff $\widehat{v}(\widehat{NNF(A)}) = 1$ (cf. Lemma 3.1 of [3]).
2. For any A and B , $A \models_{FDE} B$ iff $\widehat{NNF(A)} \models \widehat{NNF(B)}$ (cf. Theorem 3.1 of [3]).
3. For any clause D , $D \in RPI(A)$ iff $\widehat{D} \in PI(\widehat{NNF(A)})$.
4. The problem of relevant prime implicate generation is polynomially reducible to classical prime implicate generation.

PROOF. We note that (2) is a simple corollary of (1). For (1), we use an induction on the structure of $NNF(A)$. There are two base cases with either $NNF(A) = p_i$ or $NNF(A) = \neg p_i$. In the former case we have $1 \in v(p_i) \Leftrightarrow \widehat{v}(p_i^+) = 1$ given by the definition of \widehat{v} . In the later case we have $1 \in v(\neg p_i) \Leftrightarrow 0 \in v(p_i) \Leftrightarrow \widehat{v}(p_i^-) = 1$.

For the induction case we have either $NNF(A) = B \wedge C$ or $NNF(A) = B \vee C$. We note that both B and C must be in NNF form and hence the induction hypothesis applies. Thus we have $1 \in v(B) \Leftrightarrow \widehat{v}(\widehat{B}) = 1$ and

⁷The underlying 4-value assignment is based on the same ambi-valuation of Dunn [12] as deployed in the proof of Lemma 4.1. We note in passing that this is not the only semantics available for FDE .

$1 \in v(C) \Leftrightarrow \widehat{v}(\widehat{C}) = 1$. So $1 \in v(B \wedge C) \Leftrightarrow [1 \in v(B) \text{ and } 1 \in v(C)] \Leftrightarrow [\widehat{v}(\widehat{B}) = 1 \text{ and } \widehat{v}(\widehat{C}) = 1] \Leftrightarrow \widehat{v}(\widehat{B \wedge C}) = 1$. The case for $B \vee C$ is similar.

$(3 \Rightarrow)$: Since $NNF(A) \equiv_{FDE} A$ it suffices for us to consider an arbitrary $D \in RPI(\widehat{NNF(A)})$. Then by (2) above we have $\widehat{NNF(A)} \models \widehat{D}$. This shows that \widehat{D} is an implicate of $\widehat{NNF(A)}$. Toward a contradiction, suppose \widehat{D} is not prime. Then there exists a clause C such that $\widehat{NNF(A)} \models C$ and $C \models \widehat{D}$ but $\widehat{D} \not\models C$. But $\widehat{NNF(A)}$ is negation free and thus neither C nor \widehat{D} are the empty clause, nor are they tautologies. Hence there must be a C' such that $\widehat{NNF(A)} \models_{FDE} C'$ where $\widehat{C'} = C$. But then we have $C' \models_{FDE} D$ but $D \not\models_{FDE} C'$. This contradicts the primeness of D . Hence $\widehat{D} \in PI(\widehat{NNF(A)})$ as required.

$(3 \Leftarrow)$: Suppose that $D \notin RPI(\widehat{NNF(A)})$. Then either $\widehat{NNF(A)} \not\models_{FDE} D$ or D is not prime. In the former case, $\widehat{NNF(A)} \not\models \widehat{D}$ follows immediately from (2). So suppose D is a relevant implicate of $\widehat{NNF(A)}$ but is not prime. Then there exists a C such that $\widehat{NNF(A)} \models_{FDE} C$ and $C \models_{FDE} D$ but $D \not\models_{FDE} C$. By $(3 \Rightarrow)$ and (2) above it follows that \widehat{C} is a prime implicate of $\widehat{NNF(A)}$ but $\widehat{D} \not\models \widehat{C}$. Hence $\widehat{D} \notin PI(\widehat{NNF(A)})$ as required.

(4) : We note that both NNF conversion and the splitting transform are linearly related to the input formula. Hence by (3) above, the claim follows. \dashv

We note that (4) of Proposition 4.7 in fact answers a question raised by J. Marcos in WCP4—the complexity of deciding whether a given clause is a relevant prime implicate of a given formula is no harder than deciding whether a given clause is just a prime implicate of a given formula. Both decision problems are BH_2 -complete (see Marquis [24], Proposition 100).⁸

5. Conclusion

In this paper we have seen that reasoning with inconsistent information can be divided into two distinct stages. In the first stage inconsistent information encoded in a full language can be rewritten in such a way as to

⁸This result is not to be confused with the result of Umans [40], Theorem 5, which is the problem of finding the *shortest implicant*. Umans shows that the shortest implicant problem is Σ_2^2 -complete.



facilitate the isolation of the inconsistent part of the information. In the second stage various deduction strategies based on either classical or non-classical logics can then be applied to the rewrite. We note that Belnap's strategy of dividing reasoning into a *preprocessing* stage and a *deduction* stage is akin to a recent approach to *knowledge compilation*. The key to preprocessing data is to ensure that any operation on data must be *content preserving*. However, we find Belnap's suggestion of using *conjunctive containment* wanting. In particular, inconsistent information tends to interact badly with disjunctive and redundant information. Although conjunctive containment generally reduces disjunctive consequences, it is however insufficient. Our remedy is to use a relevant notion of prime implicates as the basis to both preserve information and minimise the potentially harmful disjunctive content of inconsistent information.

References

- [1] A. E. Anderson and N. D. Belnap, *Entailment: the Logic of Relevance and Necessity*, Vol 1. Princeton University Press, 1975.
- [2] A. E. Anderson, N. D. Belnap, and J. M. Dunn, *Entailment: the Logic of Relevance and Necessity*, Vol 2. Princeton University Press, 1992.
- [3] O. Arieli and M. Denecker, "Modelling paraconsistent reasoning by classical logic", pages 1–14 in: T. Eiter and K-D. Schewe, editors, *Proceedings of the Second International Symposium on Foundations of Information and Knowledge Systems (FoIKS)*, Lecture Notes in Computer Science 2284. Springer Verlag, 2002.
- [4] O. Arieli and M. Denecker, "Reducing preferential paraconsistent reasoning to classical entailment", *Journal of Logic and Computation* 13, 4 (2003): 557–580.
- [5] N. D. Belnap, "Rescher's hypothetical reasoning", pages 19–28 in: E. Sosa, editor, *The Philosophy of Nicholas Rescher: Discussion and Replies*. D. Reidel Pub., 1979.
- [6] N. D. Belnap, "Conjunctive containment", pages 145–156 in: J. Norman and R. Sylvan, editors, *Directions in Relevant Logic*. Kluwer Academic Pub., 1989.
- [7] P. Besnard and A. Hunter, "Quasi-classical logic: Non-trivializable classical reasoning from inconsistent information", pages 44–51 in: *Symbolic and Quantitative Approaches to Reasoning and Uncertainty 95*, Lecture Notes in Artificial Intelligence 946. Springer Verlag, 1995.

- [8] P. Besnard and T. H. Schaub, “Signed systems for paraconsistent reasoning”, *Journal of Automated Reasoning* 20 (1998): 191–213.
- [9] J. de Kleer, “Focusing on probable diagnoses”, pages 842–848 in: *Proceedings of the National Conference on Artificial Intelligence (AAAI 1991)*. Morgan Kaufmann, 1991.
- [10] J. de Kleer, A. K. Mackworth, and R. Reiter, “Characterizing diagnoses”, pages 324–330 in: *Proceedings of the National Conference on Artificial Intelligence (AAAI 1990)*. Morgan Kaufmann, 1990.
- [11] J. de Kleer, A. K. Mackworth, and R. Reiter, “Characterizing diagnoses and systems”, *Artificial Intelligence* 56, 2–3 (1992): 197–222.
- [12] J. M. Dunn, “Relevance logic and entailment”, pages 117–224 in: D. M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, Volume 3: Alternatives To Classical Logic. D. Reidel Pub., 1986.
- [13] K. Gemes, “A new theory of content I: Basic content”, *Journal of Philosophical Logic*, 23, 6 (1994): 595–620.
- [14] K. Gemes, “A new theory of content II: Model theory and some alternatives”, *Journal of Philosophical Logic* 26, 4 (1997): 449–476.
- [15] G. Gottlob, “Complexity results for nonmonotonic logics”, *Journal of Logic and Computation* 2 (1992): 397–425.
- [16] C. Hewitt, “Large-scale organisational computing requires unstratified reflection and strong paraconsistency”, pages 110–124 in: *Coordination, Organizations, Institutions, and Norms in Agent Systems III: COIN 2007 International Workshops*, Lecture Notes in Artificial Intelligence 4870, 2008.
- [17] C. Hewitt, “Common sense for concurrency and inconsistency tolerance using Direct LogicTM and the Actor Model”. Published online <http://arxiv.org/abs/0812.4852>
- [18] J. F. Horty, “Nonmonotonic foundations for deontic logic”, pages 17–44 in: D. Nute, editor, *Defeasible Deontic Logic*, Studies in Epistemology, Logic, Methodology, and Philosophy of Science vol 263. Kluwer Academic Pub., 1997.
- [19] A. Hunter, “Reasoning with contradictory information using quasi-classical logic”, *Journal of Logic and Computation* 10, 5 (2000): 677–703.
- [20] A. Hunter, “A semantic tableau version of first-order quasi-classical logic”, pages 544–555 in: S. Benferhat and P. Besnard, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 6th European Conference, ECSQARU 2001, Toulouse, France, September 19–21, 2001, Proceedings*, Lecture Notes in Artificial Intelligence 2143. Springer Verlag, 2001.



- [21] A. Hunter, “Measuring inconsistency in knowledge via quasi-classical models”, pages 68–73 in: *Proceedings of the 18th National Conference of Artificial Intelligence (AAAI 2002)*, 2002.
- [22] H. Kautz and B. Selman, “A general framework for knowledge compilation”, pages 287–300 in: H. Richter and M. Richter, editors, *Proceedings of International Workshop on Processing Declarative Knowledge*, Lecture Notes in Artificial Intelligence 567. Springer Verlag, 1991.
- [23] H. Kautz and B. Selman, “Knowledge compilation and theory approximation”, *Journal of the ACM* 43, 2 (1996): 193–224.
- [24] P. Marquis, “Consequence finding algorithms”, pages 41–145 in: S. Moral and J. Kohlas, editors, *Algorithms for Defeasible and Uncertain Reasoning*, volume 5 of *Handbook on Defeasible Reasoning and Uncertainty Management Systems*, chapter 2. Kluwer Academic Pub., 2000.
- [25] B. Nebel, “Belief revision and default reasoning: Syntax-based approaches”, pages 417–428 in: J. A. Allen, R. Fikes, and E. Sandewall, editors, *Principle of Knowledge Representation and Reasoning: Proceedings of the Second International Conference (KR91)*. Morgan Kaufmann, 1991.
- [26] B. Nebel, “Syntax-based approaches to belief revision”, pages 52–88 in: P. Gärdenfors, editor, *Belief Revision*. Cambridge University Press, 1992.
- [27] W. V. O. Quine, “The problem of simplifying truth functions”, *American Mathematical Monthly*, 59 (1952): 521–531.
- [28] W. V. O. Quine, “A way to simplify truth functions”, *American Mathematical Monthly* 62 (1955): 627–631.
- [29] W. V. O. Quine, “On cores and prime implicants of truth functions”, *American Mathematical Monthly* 66 (1959): 755–760.
- [30] A. Ramesh, G. Becker, and N. V. Murray, “CNF and DNF considered harmful for computing prime implicants/implicates”, *Journal of Automated Reasoning* 18 (1997): 337–356.
- [31] A. Ramesh, B. Beckert, R. Hähnle, and N. V. Murray, “Fast subsumption checks using anti-links”, *Journal of Automated Reasoning* 18 (1997): 47–83.
- [32] A. G. Ramesh, *Some Applications of Non Clausal Deduction*, PhD thesis, Department of Computer Science, State University of New York at Albany, 1995.
- [33] R. Reiter, “A logic for default reasoning”, *Artificial Intelligence* 13 (1980): 81–132.

- [34] N. Rescher, *Hypothetical Reasoning*, North-Holland, 1964.
- [35] N. Rescher, *The Coherence Theory of Truth*, Oxford University Press, 1973.
- [36] N. Rescher and R. Manor, “On inference from inconsistent premisses”, *Theory and Decision* 1 (1970): 179–217.
- [37] T. H. Schaub, “The family of default logics”, pages 77–134 in: P. Besnard and A. Hunter, editors, *Handbook of Defeasible Reasoning and Uncertain Information Volume 2, Reasoning with Actual and Potential Contradictions*. Kluwer Academic Pub., 1998.
- [38] J. Stillman, “It’s not my default: The complexity of membership problems in restricted propositional default logics”, pages 571–578 in: *Proceedings of the 8th National Conference on Artificial Intelligence, AAAI*. MIT Press, 1990.
- [39] N. Tennant, “Frege’s content-principle and relevant deducibility”, *Journal of Philosophical Logic* 32 (2003): 245–258.
- [40] Ch. Umans, “The minimum equivalent DNF problem and shortest implicants”, page 556 in: *FOCS ’98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, 1998. IEEE Computer Society.

PAUL WONG
Australian National University
ACT 0200 Canberra, Australia
paul.wong@anu.edu.au