GREG RESTALL

# How to be *Really* Contraction Free

**Abstract.** A logic is said to be *contraction free* if the rule from $A \to (A \to B)$ to $A \to B$ is not truth preserving. It is well known that a logic has to be contraction free for it to support a non-trivial naïve theory of sets or of truth. What is not so well known is that if there is *another* contracting implication expressible in the language, the logic still cannot support such a naïve theory. A logic is said to be *robustly* contraction free if there is no such operator expressible in its language. We show that a large class of finitely valued logics are each not robustly contraction free, and demonstrate that some other contraction free logics fail to be robustly contraction free. Finally, the sublogics of $\mathbf{L}_\omega$ (with the standard connectives) are shown to be robustly contraction free.

## 1. Comprehension and Implications

A *naïve comprehension scheme* is a collection of all formulæ of the form $(\exists x)(\forall y)(y \in x \leftrightarrow \phi(y))$ (where $\phi(y)$ does not have $x$ free) in some appropriate language. Let $C$ be such a set of formulæ. We are interested in the consequences of $C$, that is the formulæ $A$ such that $C \vdash A$, for some appropriate notion of deduction. One example of a theory containing a naïve comprehension scheme is naïve set theory. There are others too. For example, naïve property theory, where for every predicate there is a corresponding property of just those things of which the predicate truly predicates. We will consider an arbitrary theory containing a naïve comprehension scheme, and to make life easier, we will follow the standard set theoretic notation, and take '$\{x : \phi\}$' to be a name in our language satisfying

$$(\forall y)(y \in \{x : \phi\} \leftrightarrow \phi(y))$$

It has long been known that naïve comprehension and the rule of contraction

$$\frac{A \to (A \to B)}{A \to B}$$

*Studia Logica* **52**: 381–391, 1993.

do not go together at all well. The reason for this is well worn. Let $a = \{x : x \in x \to A\}$ for any sentence $A$. Then we have

| | | |
|---|---|---|
| 1. | $\mathcal{C} \vdash a \in a \leftrightarrow (a \in a \to A)$ | by comprehension. |
| 2. | $\mathcal{C} \vdash a \in a \to (a \in a \to A)$ | simplifying 1. |
| 3. | $\mathcal{C} \vdash a \in a \to A$ | contracting 2. |
| 4. | $\mathcal{C} \vdash (a \in a \to A) \to a \in a$ | simplifying 1. |
| 5. | $\mathcal{C} \vdash a \in a$ | *modus ponens* on 3 and 4. |
| 6. | $\mathcal{C} \vdash A$ | *modus ponens* on 3 and 5. |

$A$ is arbitrary, so if we can soundly reason like that, we can prove anything. Granting the comprehension scheme, and granting that $\mathcal{C}$ is closed under *modus ponens* for $\to$ (which you ought to grant, if you take it that your theory is anything like being *true*, and that $\to$ is a genuine conditional) the only objectionable move is contraction. To keep the hope of a non-trivial naïve comprehension scheme alive, contraction must be abandoned.

A moment's thought reveals that there is nothing particularly special about the operator $\to$ used in the definition of $a$. To trivialise our theory all we need is *some* operator $>$ in expressible in the language that has these three properties:

| | |
|---|---|
| (1) | $\dfrac{A \to B}{A > B}$ |
| (2) | $\dfrac{A > (A > B)}{A > B}$ |
| (3) | $\dfrac{A,\ A > B}{B}$ |

Table 1

Because if we have a conditional like this, setting $b = \{x : x \in x > B\}$ gives the following proof

| | | |
|---|---|---|
| 1. | $\mathcal{C} \vdash b \in b \leftrightarrow (b \in b > B)$ | by comprehension. |
| 2. | $\mathcal{C} \vdash b \in b \to (b \in b > B)$ | simplifying 1. |
| 3. | $\mathcal{C} \vdash b \in b > (b \in b > B)$ | condition **(1)** on 2. |
| 4. | $\mathcal{C} \vdash b \in b > B$ | condition **(2)** on 3. |
| 5. | $\mathcal{C} \vdash (b \in b > B) \to b \in b$ | simplifying 1. |
| 6. | $\mathcal{C} \vdash b \in b$ | *modus ponens* on 4 and 5. |
| 7. | $\mathcal{C} \vdash B$ | condition **(3)** on 4 and 6. |

Given that we call a logic without the contraction rule *contraction free*, the following definition lends itself for use:



DEFINITION 1 An operator satisfying the three conditions given in Table 1 is said to be a *contracting implication*. If a logic contains *no* contracting implication it is said to be *robustly contraction free*.

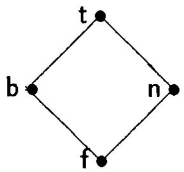
We have shown the following:

THEOREM 1 *A logic must be robustly contraction free if it is to support a nontrivial naïve comprehension scheme.*

Moh Shaw-Kwei showed in 1954 [7] that none of the $n$-valued Łukasiewicz systems are robustly contraction free, but to date, this result has not been extended to a larger class of propositional logics. In this chapter we show that *no* finitely valued logic satsifying certain general conditions is robustly contraction free, and that some other contraction-free logics also fail in this regard. Finally, we will deliniate a class of robustly contraction free logics. To start, we will consider the logic **BN4**.

## 2. BN4...

As far as finitely valued logics go, **BN4** has had a great deal of good press in the recent past [1, 4, 8] and deservedly so. It is probably the most natural four-valued logic there is. It is the logic you would be naturally led to, given that you want to extend the boolean semantics for classical logic by admitting values that are both true and false (b) and neither (n). Its b-free fragment is $\mathbf{Ł}_3$, the famous three-valued logic of Łukasiewicz, and its n-free fragment is **RM3**, the memorable almost-relevant logic. The first interpretation of the ($\to$-free part of the) logic in terms of the 'both' and 'neither' values seems to have been given by Mike Dunn in a 1968 APA-ASL symposium "Natural vs. Formal Languages", and in his 'coupled trees' paper [2] which uses it to characterise first degree entailment. However, the structure is so natural that it seems to be reinvented with monotonous regularity. A simple way to define **BN4** is by way of its four element algebra, given below:




| $\to$ | t | b | n | f |
|---|---|---|---|---|
| *t | t | f | n | f |
| *b | t | b | n | f |
| n | t | n | t | n |
| f | t | t | t | t |

| | $\sim$ |
|---|---|
| *t | f |
| *b | b |
| n | n |
| f | t |

As usual, $\wedge$ is the greates lower bound, and $\vee$ the least upper bound, so $\mathsf{f} \wedge x = \mathsf{f}$ for any $x$ and $\mathsf{n} \vee \mathsf{b} = \mathsf{t}$. Negation and implication are defined by the matrices given above, and the asterisks in the diagram show which elements are designated. So, for a sentence to count as a theorem of **BN4** it must be sent to t or b under any evaluation of the variables.

It is a pleasing exercise to show that **BN4** properly contains **C** (also called **R–W** or **RW** in the literature), and its close cousin, additive and multiplicative linear logic. That it contains these logics is simply a matter of axiom chopping — which we unashamedly leave to the reader. All that must be done is to show that the axioms are true under any assignment, and that the rules are truth preserving. Here is an axiomatisation:

1. $A \wedge B \to A,\ A \wedge B \to B$
2. $(A \to B) \wedge (A \to C) \to (A \to B \wedge C)$
3. $A \to A \vee B,\ B \to A \vee B$
4. $(A \to C) \wedge (B \to C) \to (A \vee B \to C)$
5. $A \to ((A \to B) \to B)$
6. $(A \to B) \to ((C \to A) \to (C \to B))$
7. $\sim\sim A \to A$
8. $(A \to B) \to (\sim B \to \sim A)$
9. $A, B \vdash A \wedge B$
10. $A, A \to B \vdash B$
11. $A \wedge (B \vee C) \to (A \wedge B) \vee (A \wedge C)$

**C** is given by 1–11, and the additive/multiplicative fragment of linear logic by 1–10.

To show that **BN4** properly contains **C** it is sufficitent to note that

$$(A \to B) \vee (B \to C) \vee (B \leftrightarrow \sim B)$$

It is not a theorem of **C** (as none of its disjuncts are, and **C** is prime) but it comes out designated in **BN4** every time. It is also easy to show that contraction fails in **BN4** — $\mathsf{n} \to (\mathsf{n} \to \mathsf{b}) = \mathsf{t}$ but $\mathsf{n} \to \mathsf{b} = \mathsf{n}$.

Despite the fact that **BN4** is contraction free, it fails to be robustly contraction free. To see this, define a connective $>$ evaluated by the condition that $x > y$ is $x \wedge \mathsf{b} \to y$. It is easy to see that this connective satisfies conditions **(1)** and **(3)**. We are left with verifying **(2)**. But this is simple, given the table:

| > | t | b | n | f |
|---|---|---|---|---|
| t | t | b | n | f |
| b | t | b | n | f |
| n | t | t | t | t |
| f | t | t | t | t |

For $x > y$ to be undesignated, we must have $x$ as either t or b, and $y$ as either n or f. And in this case $x > y = y$. It follows that $x > (x > y) = x > y$, so $x > (x > y)$ is undesignated also. Contraposing this we see that $>$ contracts.

## 3. ... Finitely Valued Logics ...

As indicated before, the trouble with **BN4** is not restricted, but rather, it is suggestive of a problem that plagues all finitely valued logics. The



problem can be explained like this. Given an implication functor $>$, we can define another implication functor in terms of it, but weaker — either as $x \wedge \tau > y$, for some true constant $\tau$, or as $x > (x > y)$, or by some other means. However we do it, the new operator will satisfy *modus ponens*, and it will be weaker that $>$, in some sense to be explained. We continue this weakening process *ad infinitum*, and eventually — finitely valued logics being the cramped places that they are — we shouldn't get anything new. Once this happens, we get contraction. This is the guiding idea in what follows.

Definition 2 A set $V$ will define a *semilattice logic* if and only if it satisfies the following conditions:

- There is an operator $\wedge$ on $V$ that defines a semilattice ordering $\leq$ on $V$. In other words, $\wedge$ is idempotent, symmetric, and associative, and $x \leq y$ iff $x \wedge y = x$.
- There is a set $D$ of *designated* elements in $V$. $D$ forms a filter. In other words, if $x \in D$ and $x \leq y$ then $y \in D$, and if $x, y \in D$ then $x \wedge y \in D$.
- The conjunction of all elements of $D$ (itself an element of $D$) can be *named* in the language. We will call it $t$. $t$ is the smallest element of $D$. (In linear logic, this constant is the multiplicative identity **1**.)
- There is an operator $\to$ that satisfies $x \leq y$ iff $t \leq x \to y$.

*Some* weak logics are outside the scope of this definition, but they are rare, and typically, not finitely valued.

We will consider an arbitrary finite semilattice logic, and show that it can express a contracting implication. The first result is a lemma concerning the behaviour of $\to$.

Lemma 2 (Prefixing) *If* $t \leq x \to y$ *then* $t \leq x \wedge z \to y$.

Proof. If $t \leq x \to y$ then $x \leq y$ and so $x \wedge z \leq y$ by the properties of semilattices. This gives $t \leq x \wedge z \to y$. ■

We define an infinite family of operators:

Definition 3 For each $n = 0, 1, 2, \ldots$ define $>_n$ on $V$ by fixing

$$x >_0 y = y \qquad x >_{n+1} y = x \wedge t \to (x >_n y)$$

It is easy to show that for each $m, n$, $x >_m (x >_n y) = x >_{m+n} y$.

**Lemma 3** *If $t \leq y$ then $t \leq x >_1 y$.*

**Proof.** If $t \leq y$ then $t \leq t \to y$, and hence $t \leq x \wedge t \to y$ by lemma 2. ■

**Lemma 4** *If $t \leq x >_n y$ then $t \leq x >_{n+1} y$, for each $n = 0, 1, 2, \ldots$*

**Proof.** This follows from lemma 3, as $x >_{n+1} y = x >_1 (x >_n y)$. ■

**Lemma 5** *If $t \leq x \to y$ then $t \leq x >_n y$ for each $n > 0$.*

**Proof.** If $t \leq x \to y$ then $t \leq x \wedge t \to y = x >_1 y$. Then $n - 1$ applications of the previous lemma gives us the result. ■

In other words, each $>_n$ satisfies condition **(3)**. Furthermore, we can show that $>_n$ satisfies condition **(1)**.

**Lemma 6** *If $t \leq x >_n y$ and $t \leq x$ we also have $t \leq y$.*

**Proof.** Clearly this result holds for $n = 0$. If it holds for $n$, note that $t \leq x >_{n+1} y = x \wedge t \to (x >_n y)$, so *modus ponens* for $\to$ (using $t \leq x \wedge t$) gives $t \leq x >_n y$, and our hypothesis gives us the result.[1] ■

Each of these results work in *any* semilattice logic. The finiteness condition is used in the following:

**Lemma 7** *For some $n$, $t \leq x >_n y$ if and only if $t \leq x >_{n'} y$ for each $n' > n$.*

**Proof.** For convenience, name the $m$ elements of $V$ as $x_1, \ldots, x_m$. For each implication $>_n$, consider the $m \times m$ matrix $A_n$, where the $(i, j)$ element is 1 if and only if $x_i >_n x_j$ is designated, and is 0 otherwise. In other words, $[A_m]_{ij} = 1$ if and only if $t \leq x_i >_n x_j$, and $[A_m]_{ij} = 0$ otherwise. By lemma 4, the sequence $A_0, A_1, A_2, \ldots$ is a monotonic sequence of matrices, in that for each $i, j$, once the $(i, j)$ element gets the value 1, it keeps that value in every matrix in the sequence. It follows that for some $n$, $A_{n'} = A_n$ for each $n' > n$. To see this, take $n_{ij}$ to be the least $n$ where $[A_n]_{ij} = 1$, or let it be 0 if there is no such $n$. Take $n = \max(n_{ij})$, this is the desired number.

But if $A_{n'} = A_n$ for each $n' > n$, it follows that $t \leq x >_n y$ if and only if $t \leq x >_{n'} y$ for each $n' > n$, as desired. ■

Given this, we can prove our major theorem.

**Theorem 8** *Any finitely valued semilattice logic is not robustly contraction free.*

**Proof.** Let $n$ be such that $t \leq x >_n y$ if and only if $t \leq x >_{n'} y$ for each $n' > n$. Then $t \leq x >_n y$ if and only if $t \leq x >_{2n} y$. But this is simply

[1] For those interested in these matters, the proof of this lemma used a contraction in the metalanguage, as the premise to the effect that $t \leq x$ was discharged twice.



$t \leq x >_n (x >_n y)$, and so, $>_n$ satisfies condition **(2)**. But by previous lemmas $>_n$ also satisfies conditions **(1)** and **(3)**. ■

It follows that *any* finitely valued semilattice logic is useless for the project of formalising a nontrivial naïve comprehension scheme.

## 4. ... and Others

It is interesting to note that other contraction free logics also fail to be robustly contraction free, despite the fact that they are not finitely valued. The logic **BN**, introduced in Slaney, Surendonk and Girle's "Time, Truth and Logic" [9], is given by adding two axioms to **C**. We need $>_1$ and $>_2$ defined as before, and then **BN** is given by adding:

$$A \circ B >_2 B \qquad (A >_2 B) \wedge ({\sim} B >_2 {\sim} A) \to (A \to B)$$

It is simple to show that $>_2$ is a contracting implication in **BN**, and that so, it fails to be robustly contraction free.

More interestingly, Abelian Logic (called **A**) studied by Meyer and Slaney [5] also fails to be robustly contraction free.[2] **A** can be defined by way of a particular propositional structure — on the set $Z$ of integers. The lattice ordering is the obvious one on $Z$, and so, conjunction and disjunction are min and max respectively. The implication $x \to y$ is simply $y - x$, and so, the designated values are the non-negative integers. Any number of negations can be defined, but the canonical example is given by taking ${\sim} x$ to be $x \to 0$, which is simply $-x$. A simple check verifies that **A** properly contains **C**. In fact, it is given by adding to **C** the axiom:

$$((A \to B) \to B) \to A$$

which can be seen as a generalised double negation axiom. One interesting fact about **A** is that it has *no* nontrivial finite propositional structures as models. (To show this, note that the presence of an object $F$ such that $F \leq x$ for each $x$ in a structure will lead to triviality. This is because $(x \to F) \to F$ is equal to $F \to F$, which is $T$ (the top element, ${\sim} F$) but $(x \to F) \to F \leq x$, so $T \leq x$.)

[2] This result is due to Graham Priest, Paul Pritchard and myself, with some assistance to the Automated Reasoning Project at the Australian National University, who brought us together for the weekend, and Ansett Australia, who ensured that we would have a two hour wait at Sydney Airport, with nothing much better to do than a bit of number theory. I'm grateful to Graham, Paul and the people of the ARP for their interest and help.

It is also a simple exercise to show that $>_n$ will not contract for each $n = 1, 2, \ldots$ So, to show that **A** is not robustly contraction free, we must take another approach. We must find a function $f : Z^2 \to Z$, satisfying:

| | |
|---|---|
| **(1a)** | If $x \leq y$ then $f(x,y) \geq 0$ |
| **(2a)** | If $f(x, f(x,y)) \geq 0$ then $f(x,y) \geq 0$ |
| **(3a)** | If $f(x,y) \geq 0$ and $x \geq 0$ then $y \geq 0$ |

Table 2

where $f$ has been defined in terms of addition, subtraction, min, max and zero. The associated logical operator with then satisfy the conditions required for a contracting implication.

The first thing to note is that a function $f$ satisfying $f(x,y) < 0$ iff $x > 0$ and $y < 0$ will satisfy these conditions. (This is a simple verification, and is left to the reader.) The challenge is to define such a function in terms that are allowed. One such function is given as follows:

$$f(x,y) = \max(y,0) + \max(-x,0) + \min(\max(-x,y),0)$$

That it works is left to the reader. The corresponding implication operator is the horrible looking:

$$A > B = (B \vee 0) \circ ({\sim} A \vee 0) \circ (({\sim} A \vee B) \wedge 0)$$

This 'simplifies' to

$$A > B = {\sim}(({\sim} A \vee B) \wedge 0 \to ((B \vee 0) \to (A \wedge 0)))$$

It follows that while **A** is contraction free in that no simple minded nesting of arrows will yield a contracting implication, a more devious approach will give one. This failure of **A** to be robustly contraction free is somewhat surprising, for it explicity contains an infinite number of truth-values, which ought to be enough to distinguish any number of repetitions of premises. What strikes this observer is that some form of disjunctive syllogism might be lurking in **A**, allowing the operation $>$ to detatch. (After all, $A > B$ does contain ${\sim} A \vee B$ as a subformula.) However, this is all speculation.

## 5. A Generalisation

Curry paradoxes are not just given by operators $>$ that satisfy contraction in the sense of condition **(2)**.[3] If we have an operator $>$ that satisfies **(1)**,

[3] The idea for this section came from the referee, whose comments were most helpful.



**(3)** and the rule from $A > (A > (A > B))$ to $A > (A > B)$, then a naïve comprehension scheme is also trivialised. This could be called 3–2 contraction, instead of the 2–1 contraction of condition **(2)**. In general, an operator satisfying **(1)**, **(3)** and $(N+1)$–$N$ contraction is enough to trivialise a naïve comprehension scheme. It might be thought that this gives us a new way of trivialising the scheme, but this is not the case. We will show that this doesn't add anything new, as any 2–1 contraction free logic is also $(N+1)$–$N$ contraction free. To do this, we expand our definition a little.

DEFINITION 4 Let $A >^0 B$ be $B$ and let $A >^{n+1} B$ be $A > (A >^n B)$. (Note that this definition differs from the previous kind of iterated implication, in that $t$ is absent.) Clearly $A >^{N+M} B = A >^N (A >^M B)$. The implication $>$ is an *$M$–$N$ contracting operator* (where $M > N$) if the rule from $A >^M B$ to $A >^N B$ is truth preserving.

This is enough for our preliminary result.

LEMMA 9 *If $>$ is an $(N+1)$–$N$ contracting operator then it is also an $M$-$N$ contracting operator for each $M > N$.*

PROOF. Prove the result by induction on $M$. The base case, $M = N+1$ holds by definition. Suppose that we have $A >^M B \vdash A >^N B$, for $M > N$. Then $A >^{M+1} B = A >^{N+1} (A >^{M-N} B) \vdash A >^N (A >^{M-N} B) = A >^M B$, by the fact that $>$ is $(N+1)$–$N$ contracting, and by the induction hypothesis, this gives $A >^N B$, as desired. ■

LEMMA 10 *If $>$ is an $(N+1)$–$N$ contracting operator in a logic, then $>^N$ is a 2–1 contracting operator in that logic.*

PROOF. By the lemma, $>$ is $2N$–$N$ contracting, so we have $A >^{2N} B \vdash A >^N B$, so $A >^N (A >^N B) \vdash A >^N B$ as desired. ■

So, if we have an $(N+1)$–$N$ contracting operator that also satisfies **(1)** and **(3)**, we have an associated operator that satisfies **(2)**, and also **(3)** (as can easily be checked). What is not so obvious is that it will satisfy **(1)**. If $>^{2N}$ does not satisfy **(1)** it is easy to define an operator that does. Take $\Rightarrow$ to be given by setting $A \Rightarrow B = A \wedge t >^{2N} B$. By lemmas 2 to 6, this operator will satisfy **(1)** and **(3)**, and **(2)** follows from the validity of contraction for $>^{2N}$. So, we have the following result:

THEOREM 11 *If a semilattice logic is 2–1 contraction free, it is also $(N+1)$–$N$ contraction free.*

## 6. The Trouble Avoided

Thankfully it is clear that many logics *are* robustly contraction free. One such is Łukasiewicz's infinitely valued logic. In it there are no contracting implication operators, as can be seen by the fact that the naïve comprehension scheme is consistent in that logic [10]. It follows that any of its sublogics are robustly contraction free.

We will end this paper with a conjecture. It does not look particularly easy to prove, but any headway made on it would be most welcome.

CONJECTURE *A logic is robustly contraction free if and only if it nontrivially supports a naïve comprehension scheme.*

Obviously this will only work given that the logic satisfies a few small conditions — namely that it can express the comprehension scheme. We have shown that robust contraction freedom is *necessary* for nontriviality. The hard part is proving sufficiency. This is where you, gentle reader, have your turn.[4]

## References

[1] N. D. BELNAP, *A Useful Four-Valued Logic,* in J. M. Dunn and G. Epstein (eds.) Modern Uses of Multiple–Valued Logics, D. Reidel Publ. Co., Dordrecht .

[2] J. M. DUNN, *Intuitive Semantics for First-Degree Entailments and 'Coupled Trees',* **Philosophical Studies 29** (1976), pp. 139–168.

[3] J. M. DUNN, and G. EPSTEIN, (eds) **Modern Uses of Multiple-Valued Logics**, D. Reidel Publ. Co., Dordrecht 1977.

[4] R. K. MEYER, S. GIAMBRONE and R. T. BRADY, *Where Gamma Fails,* **Studia Logica 43** (1984), pp. 247–256.

[5] R. K. MEYER and J. K. SLANEY, *Abelian Logic (from A to Z),* in G.Priest, R. Routley and J. Norman (eds) **Paraconsistent Logic: Essays on the Inconsistent,** Philosophia Verlag, München, 1989.

[6] G. PRIEST, R. ROUTLEY and J. NORMAN (eds), **Paraconsistent Logic: Essays on the Inconsistent,** Philosophia Verlag, München, 1989.

[7] M. SHAW–KWEI, *Logical Paradoxes for Many-Valued Systems,* **Journal of Symbolic Logic 19** (1954), pp. 37–40.

[8] J. K. SLANEY, *The Implications of Paraconsistency,* **Proceedings of IJCAI-91, Vol. 2,** (1991), pp. 1052–1057.

[4] As well as the people mentioned above, I would like to thank the Logic Group at the University of Queensland for their helpful comments on an earlier draft of this paper.



[9] J. K. SLANEY, T. SURENDONK and R. GIRLE, *Time, Truth and Logic,* Australian National University, Automated Reasoning Project Technical Report, TR-ARP-11/89, 1989.

[10] R. B. WHITE, *The Consistency of the Axiom of Comprehension in the Infinite-Valued Predicate Logic of Łukasiewicz,* **Journal of Philosophical Logic 8** (1979), pp. 509–534.

PHILOSOPHY DEPARTMENT
THE UNIVERSITY OF QUEENSLAND
QUEENSLAND, 4072, AUSTRALIA

*Received* July 8, 1991
*Revised* June 28, 1992