Applying Saturation-Based Theorem Proving to Open Problems in Positive Implicational Logic

Branden Fitelson* & Nicolas Peltier†

March 16, 2026

Abstract

We revisit a longstanding question about the shortest single axioms for positive implicational logic. Meredith discovered several 17-symbol single axioms and asked whether shorter ones exist. Later work reduced the problem to four candidate formulas of length 15. Using the automated theorem prover Vampire, we compute saturated clause sets that yield countermodels showing that three of these candidates are not single axioms. This demonstrates the effectiveness of saturation-based reasoning for such problems, which have traditionally been studied via finite model searches.

1 Introduction

In the 1950s, and entirely by hand, Carew Meredith discovered many of the shortest single axioms for various logical systems. For instance, he reported [4] the following 17-symbol single axiom (here, written in Polish notation) for positive (*viz.*, intuitionistic) implication—assuming *modus ponens* as the sole rule of inference.[1]

$$CCCpqrCsCCqCrtCqt$$

In TPTP format (see [6] or https://www.tptp.org/), which we will now adopt for the remainder of the paper, this axiom can be rendered as follows (where 'p' is a *provable/theorem* predicate, and 'i' is implication, or what was 'C' in Polish).

```
fof(m1,axiom, ![X,Y,Z,U,V]: p(i(i(i(X,Y),Z),i(U,i(i(Y,i(Z,V)),i(Y,V)))))).
```

Modus ponens can be rendered as follows.

```
fof(mp,axiom, ![X,Y]: ((p(i(X,Y)) & p(X)) => p(Y))).
```

*Northeastern University, Boston, MA, USA

†Univ. Grenoble Alpes, CNRS, LIG, F-38000 Grenoble France

[1] Meredith [3] and Ulrich [7] later reported six other single axioms of length 17.

Meredith said he did not know whether his 17-symbol axiom was the shortest possible. Dolph Ulrich [7, 8] was able to eliminate all candidate single axioms of length 15, except the following four.

```
fof(u1,axiom, ![X,Y,Z,U]: p(i(i(i(X, Y), Z), i(i(Z, i(Z, U)), i(Y, U))))).
fof(u2,axiom, ![X,Y,Z,U]: p(i(i(X, i(Y, Z)), i(i(i(U, X), Y), i(X, Z))))).
fof(u3,axiom, ![X,Y,Z,U]: p(i(i(X, Y), i(i(i(Z, X), i(Y, U)), i(X, U))))).
fof(u4,axiom, ![X,Y,Z,U]: p(i(i(i(X, Y), Z), i(i(Y, i(Z, U)), i(Y, U))))).
```

If these four candidates could be shown not to be single axioms, then it would follow that Meredith's was the shortest possible. Various finite model searches have been performed over the years, yielding no finite counter-models. Proof searches suggest that these candidates are not axioms (since hyper-resolution and condensed detachment produce increasingly long formulæ). In this note we explain how we have been able to rule out the first three candidates, using `Vampire` [1] to find saturated sets of clauses. As far as we know, this is the first time saturation-based methods have been used to settle such open questions in sentential logic (finite models/matrices are typically used for this purpose).

2 Using Saturation to Refute the Single Axiomhood of `u1`

The following law of reflexivity is a simple theorem of positive implication.

```
fof(refl,conjecture, ![X]: p(i(X,X))).
```

When we input the three formulæ `mp`, `u1`, and `refl` into `Vampire` 5.0.0 (in `casc_sat` mode), it terminates and outputs the following saturated set of clauses.[2]

```
cnf(u42,axiom, ~p(i(X0,i(X0,X1)))).
cnf(u30,axiom, p(i(i(X2,i(X2,X3)),i(X1,X3))) | ~p(i(i(X0,X1),X2))).
cnf(u12,axiom, p(i(i(i(X0,X1),X2),i(i(X2,i(X2,X3)),i(X1,X3))))).
cnf(u52,axiom, ~p(i(i(X0,i(X1,i(X1,X2))),X1))).
cnf(u11,negated_conjecture, ~p(i(sK0,sK0))).
cnf(u10,axiom, ~p(i(X0,X1)) | ~p(X0) | p(X1)).
```

The superposition strategy used to ensure saturation consists of reversing the polarity of the symbol `p`—that is, uniformly replacing each occurrence of `p(t)` with `~p(t)`, and vice versa—and then selecting the largest maximal negative literal in the clause, if any. Any standard term order can be used. This strategy is somewhat non-trivial, and it happens to be among those employed by `Vampire` in its portfolio mode. It follows from the refutational completeness of the superposition calculus that a counter-model to the deducibility of reflexivity from `u1` + `mp` exists, and can be constructed and described from this `Vampire` output. Here is a detailed description of the model, along with an intuitive explanation of why it suffices to show that {`mp`, `u1`} $\nvdash$ `refl`. This development provides an

[2] We independently verified that this clause set is saturated, using `iProver` v0.8.1 [2].

independent, self-contained mathematical proof that u1 is not a single axiom for intuitionistic implication. Note that in the above clauses, u10 is mp, u12 is u1, and u11 is the denial of reflexivity. The domain of quantification is the set of terms built from the signature {i, sK0}, *i.e.,* the set of binary trees, with the usual interpretations of i and sK0 (i.e., sK0 is interpreted as itself and i is interpreted as the function mapping (x, y) to the term i(x, y)). The interpretation of p is defined as follows. For any term X, the truth value of p(X) is defined by induction on X; it is set to false *iff* at least one of the following four conditions holds.

1. X is an instance of i(X0, i(X0, X1)),
2. X is an instance of i(i(X0, i(X1, i(X1, X2))), X1),
3. X = i(sK0, sK0),
4. X = i(X0, X1) where p(X0) is true and p(X1) is false.

By construction, this interpretation satisfies clauses u42, u52, and u11. Clause u10 is also satisfied: if p(i(X0, X1)) is true, then either p(X0) is false or p(X1) is true (by condition 4). Let us now verify that the interpretation also satisfies u30 and u12. For u30, suppose that the clause

```
p(i(i(T2, i(T2, T3)), i(T1, T3))) | ~p(i(i(T0, T1), T2))
```

is false. Then p(i(i(T2, i(T2, T3)), i(T1, T3))) is false and p(i(i(T0, T1), T2)) is true. Hence i(i(T2, i(T2, T3)), i(T1, T3)) must satisfy one of the four conditions above. We consider the possibilities.

1. If i(i(T2, i(T2, T3)), i(T1, T3)) is an instance of i(X0, i(X0, X1)), then necessarily T1 = i(T2, i(T2, T3)), with X0 = T1 and X1 = T3. But then p(i(i(T0, i(T2, i(T2, T3))), T2)) would be true, which contradicts the definition of the semantics of p (condition 2).
2. The term i(i(T2, i(T2, T3)), i(T1, T3)) cannot be an instance of the term i(i(X0, i(X1, i(X1, X2))), X1), since the two terms are not unifiable (as we would have X1 = i(T1, T3) and T3 = i(X1, X2)).
3. The term i(i(T2, i(T2, T3)), i(T1, T3)) cannot be equal to i(sK0, sK0).
4. It cannot have the form i(X0, X1) either, with p(X0) true and p(X1) false, since p(i(T2, i(T2, T3))) cannot be true (by condition 1).

Hence, u30 is satisfied. For u12, suppose that the clause

```
p(i(i(i(T0, T1), T2), i(i(T2, i(T2, T3)), i(T1, T3))))
```

is false. Since this term is not unifiable with any of the three i(X0, i(X0, X1)), i(i(X0, i(X1, i(X1, X2))), X1), or i(sK0, sK0), condition 4 applies. Therefore, p(i(i(T0, T1), T2)) is true and p(i(i(T2, i(T2, T3)), i(T1, T3))) is false. But then the clause

```
p(i(i(T2, i(T2, T3)), i(T1, T3))) | ~p(i(i(T0, T1), T2))
```

would be false, contradicting the validity of `u30`. Thus `u12` is satisfied. This provides a mathematical definition of an infinite model, sufficient to establish satisfiability of the set of clauses {`mp`, `u1`, `u11`}, which means that `u1` is not a single axiom for positive implication.

3 Saturated Clause Sets for `u2` and `u3`

The same technique can be used to refute the single axiom-hood of Ulrich's `u2` and `u3`. The following (rather more complex) set of saturated clauses is produced by `Vampire` for the input set {`mp`, `u2`, `refl`}.[3]

```
cnf(u23,axiom, ~p(i(i(i(X0,X0),X1),i(i(i(i(X0,X0),X1),X0),X1),X2))) | p(i(i(i(X0,X0),X1),X2))).

cnf(u20,axiom, ~p(i(i(X0,i(X1,X2)),i(X0,X1))) | p(i(i(X0,i(X1,X2)),i(X0,X2)))).

cnf(u13,axiom, p(i(i(i(X3,X0),X1),i(X0,X2))) | ~p(i(X0,i(X1,X2)))).

cnf(u25,axiom, ~p(i(i(i(i(X0,X0),X1),X0),i(i(X0,X0),X1))) | p(i(i(i(i(X0,X0),X1),X0),X1))).

cnf(u12,axiom, p(i(i(X0,i(X1,X2)),i(i(i(X3,X0),X1),i(X0,X2))))).

cnf(u22,axiom, p(i(i(i(i(i(X0,X1),i(X0,X1)),X0),i(X0,X1)),i(i(i(i(X0,X1),i(X0,X1)),X0),X1)))).

cnf(u15,axiom, ~p(i(i(X0,X1),i(i(i(i(X2,X3),X0),i(X3,X1)),X4))) | p(i(i(X0,X1),X4))).

cnf(u17,axiom, ~p(i(i(X3,X0),i(i(X0,i(X1,X2)),X1))) | p(i(i(X0,i(X1,X2)),i(X0,X2)))).

cnf(u14,axiom, ~p(i(i(X3,X0),X1)) | ~p(i(X0,i(X1,X2))) | p(i(X0,X2))).

cnf(u16,axiom, ~p(i(X0,i(i(X1,X2),X3))) | p(i(X0,X3)) | ~p(i(X1,i(X0,X2)))).

cnf(u26,axiom, ~p(i(i(X0,X0),i(X0,i(X0,X0)))) | p(i(i(i(i(X0,X0),i(X0,X0)),X0),i(X0,X0)))).

cnf(u19,axiom, p(i(i(i(i(i(X0,X0),X1),X0),i(i(X0,X0),X1)),i(i(i(i(X0,X0),X1),X0),X1)))).

cnf(u18,axiom, ~p(i(X1,i(i(i(X0,i(X1,X2)),X3),X2))) | p(i(i(i(X0,i(X1,X2)),X3),X4)) | ~p(i(i(X1,X2),i(X3,X4)))).

cnf(u11,negated_conjecture, ~p(i(sK0,sK0))).

cnf(u29,axiom,

~p(i(i(i(X0,X1),i(X0,X1)),i(i(i(i(X0,X1),i(X0,X1)),X0),i(X0,X1)),X0))) | p(i(i(i(i(X0,X1),i(X0,X1)),X0),i(X0,X1)),X1))).

cnf(u10,axiom, ~p(i(X0,X1)) | ~p(X0) | p(X1)).

cnf(u28,axiom, ~p(i(i(X0,X1),i(i(i(i(X0,X1),i(X0,X1)),X0),X1),X2))) | p(i(i(X0,X1),X2))).

cnf(u30,axiom, ~p(i(i(i(i(X0,X1),i(X0,X1)),X0),i(X0,X1))) | p(i(i(i(i(X0,X1),i(X0,X1)),X0),X1))). cnf(u24,axiom,

~p(i(i(i(X0,X0),X1),i(i(i(i(X0,X0),X1),X0),i(i(X0,X0),X1)),X0))) | p(i(i(i(i(i(X0,X0),X1),X0),i(i(X0,X0),X1)),X1))).
```

The following (even more complex) set of saturated clauses is produced by `Vampire` for the input set {`mp`, `u3`, `refl`}[3].

```
cnf(u61,axiom, ~p(i(X0,i(i(X1,X2),i(X0,X3))))).

cnf(u166,axiom, ~p(i(i(X0,X1),i(i(X2,X1),i(X2,X1))))).

cnf(u196,axiom, ~p(i(i(X0,X1),i(i(X0,X1),X1)))).

cnf(u223,axiom, ~p(i(i(X0,X0),i(X1,X0)))).

cnf(u226,axiom, ~p(i(i(i(X0,X1),i(X0,X1)),X1))).

cnf(u240,axiom, ~p(i(i(X0,X1),i(X0,X1)))).

cnf(u357,axiom, ~p(i(X1,X1))).
```

---

[3] We independently verified that this clause set is saturated, using `E` v3.2.5 [5].

```
cnf(u77,axiom, ~p(i(i(X1,X3),i(X0,X1))) | p(i(i(i(X0,X1),i(X2,X3)),i(X2,X3))) | ~p(i(X1,X2))).

cnf(u14,axiom, p(i(i(X0,X1),i(i(i(X2,X0),i(X1,X3)),i(X0,X3))))).

cnf(u57,axiom, ~p(i(X0,i(i(X1,X2),i(X0,X3)))) | p(i(X0,i(X2,X3)))).

cnf(u133,axiom, ~p(i(i(X1,i(X0,X1)),i(X2,i(X3,i(X0,X1))))) | ~p(i(X1,X3)) | ~p(i(i(X0,X1),X2)) | p(i(X2,i(X3,i(X0,X1))))).

cnf(u58,axiom, ~p(i(i(X0,X1),X2)) | p(i(i(X0,X1),X1)) | ~p(i(X2,X0))).

cnf(u100,axiom, ~p(i(i(X0,X1),i(i(X0,X1),X1))) | p(i(i(i(i(X0,X1),X1),X0),X0))).

cnf(u30,axiom, p(i(i(i(X2,X0),i(X1,X3)),i(X0,X3))) | ~p(i(X0,X1))).

cnf(u12,axiom, ~p(i(X0,X1)) | ~p(X0) | p(X1)).

cnf(u130,axiom, ~p(i(i(X2,i(X3,i(X0,X1))),X2)) | ~p(i(X1,X3)) | ~p(i(i(X0,X1),X2)) | p(i(i(X2,i(X3,i(X0,X1))),i(X3,i(X0,X1))))).

cnf(u108,axiom, p(i(i(i(X0,i(X1,X0)),i(X2,i(X3,i(X1,X0)))),i(X2,i(X3,i(X1,X0))))) | ~p(i(i(X1,X0),X2)) | ~p(i(X0,X3))).

cnf(u13,negated_conjecture, ~p(i(sK0,sK0))).

cnf(u34,axiom, ~p(i(i(X2,X0),i(X1,X3))) | ~p(i(X0,X1)) | p(i(X0,X3))).

cnf(u131,axiom, p(i(i(i(X2,i(X1,i(X0,X1))),i(X3,i(X2,i(X1,i(X0,X1))))),i(X3,i(X2,i(X1,i(X0,X1)))))) |
               ~p(i(X1,X1)) | ~p(i(i(X0,X1),X2)) | ~p(i(i(X1,i(X0,X1)),X3))).

cnf(u76,axiom, ~p(i(i(i(i(X2,X0),i(X1,X3)),i(X0,X3)),X0)) | p(i(i(X0,X1),X1))).
```

Rather than trying to articulate the models implied by these saturated clause sets, we simply present the sets here, and rely on the refutational completeness of the provers `Vampire` and `E` to establish the non-axiom-hood of `u2` and `u3`.[4]

4 The Question of Ulrich's `u4`

We have been unable to use saturation-based methods (or any other techniques) to refute the single axiom-hood of `u4` (although we strongly suspect that it is not a single axiom). As a result, the question of `u4`'s status as a single axiom for positive implication remains open. What we can say is that the shortest single axioms for positive implication have length 17 *iff* `u4` is not a single axiom for positive implication.[5]

References

[1] Bártek, F., Bhayat, A., Coutelier, R., Hajdu, M., Hetzenberger, M., Hozzová, P., Kovács, L., Rath, J., Rawson, M., Reger, G., Suda, M., Schoisswohl, J., & Voronkov, A. The Vampire Diary. In 37th International Conference on Computer Aided Verification. LNCS 15933, pp 57-71. Springer 2025. `https://doi.org/10.48550/arxiv.2506.03030`

[4] We have created a repository ⟨`https://doi.org/10.5281/zenodo.18747868`⟩ containing three `TPTP` input files: `u1.p`, `u2.p`, and `u3.p`, which will generate the saturated sets appearing in the paper. The files are self-contained — they include both descriptions and instructions for running them using `Vampire 5.0.1`.

[5] With the assistance of Matthew Walsh, we have independently verified that all of the other 15-symbol intuitionistic implicational theorems are not single axioms.

[2] K. Korovin, iProver — An Instantiation-Based Theorem Prover for First-Order Logic (System Description). In IJCAR 2008, LNCS 5195, pp. 292–298. Springer, 2008.

[3] A.N. Prior, and C. Meredith, Notes on the axiomatics of the propositional calculus, Notre Dame J. Formal Logic 4 (1963), 171–187.

[4] C. Meredith, A single axiom of positive logic. The journal of computing systems, vol. 1 no. 3 (1953), pp. 169–170.

[5] S. Schulz, S. Cruanes, and P. Vukmirović, Faster, higher, stronger: E 2.3. In International Conference on Automated Deduction (pp. 495-507). Cham: Springer International Publishing.

[6] G. Sutcliffe, Stepping Stones in the TPTP World. In IJCAR, LNCS 14739, pp 30-50. Springer, 2024.

[7] D. Ulrich, New single axioms for positive implication, Bull. Sect. Logic 28 (1999), 39–42.

[8] D. Ulrich, A legacy recalled and a tradition continued. Journal of Automated Reasoning, 27(2), (2001) 97-122.