Intuitive Counterexamples for Constructive Fallacies

James Lipton\textsuperscript{1} and Michael J. O'Donnell\textsuperscript{2}

\textsuperscript{1} Wesleyan University
\textsuperscript{2} The University of Chicago

Abstract. Formal countermodels may be used to justify the unprovability of formulae in the Heyting calculus (the best accepted formal system for constructive reasoning), on the grounds that unprovable formulae are not constructively valid. We argue that the intuitive impact of such countermodels becomes more transparent and convincing as we move from Kripke/Beth models based on possible worlds, to Läuchli realizability models. We introduce a new semantics for constructive reasoning, called relational realizability, which strengthens further the intuitive impact of Läuchli realizability. But, none of these model theories provides countermodels with the compelling impact of classical truth-table countermodels for classically unprovable formulae. We outline a proof that the Heyting calculus is sound for relational realizability, and conjecture that there is a constructive choice-free proof of completeness. In this respect, relational realizability improves the metamathematical constructivity of Läuchli realizability (which uses choice in two crucial ways to prove completeness) in the same sort of way Beth semantics improves Kripke semantics.

1 The Intuitive Impact of Countermodels

Imagine that we believe in the suitability of a formal system $\mathcal{P}$ of proof as a basis for useful reasoning in some language $\mathcal{L}$. We are trying to sell $\mathcal{P}$ to a customer. The customer is not trained in formal metamathematics, but she has an excellent intuitive grasp of the meanings of given formulae in concrete circumstances. There are a number of grounds on which the customer might challenge the suitability of $\mathcal{P}$, but we focus attention on one. Suppose that she produces a formula $\alpha$ that is not provable in $\mathcal{P}$, and argues that therefore $\mathcal{P}$ is not powerful enough for practical use. We must convince her, on intuitive grounds, that the failure of $\mathcal{P}$ to prove $\alpha$ is a useful feature, rather than a serious lacuna.

To this end, we must describe a conceivable set of circumstances $\mathcal{C}$ that might occur in the real world that language $\mathcal{L}$ describes, and interpret the primitive nonlogical symbols of $\alpha$ in $\mathcal{C}$ in such a way that $\alpha$ is clearly false. So, we argue, $\mathcal{P}$ is excused from proving $\alpha$ because $\alpha$ is not valid (not reliably true in all circumstances).

In order to prepare for such a debate, we might study formal semantics. In particular, we formalize the notion of a "conceivable set of circumstances" as a model for $\mathcal{L}$, and define what it means for a formula $\alpha$ of $\mathcal{L}$ to be true
in a particular model \( \mathcal{M} \). Then, we formalize validity of a formula as truth in all models. Finally, we demonstrate that \( \mathcal{P} \) is sound (every provable formula is valid) and complete (every valid formula is provable) with respect to our formal model theory. Soundness may help to answer a challenge of incorrect proof, while completeness may help to answer the challenge of insufficient power proposed above.

Back to the customer, who has criticized \( \mathcal{P} \) for its inability to prove \( \alpha \). Our demonstration of completeness, if it is appropriately constructive, provides us with a countermodel \( \mathcal{M}_C \) in which \( \alpha \) is false. In order to convince the customer, we must translate \( \mathcal{M}_C \) into an intuitive description of conceivable real circumstances in which \( \alpha \) is clearly false. Of course the success of our argument depends on the intuitive beliefs that our customer holds. If the customer’s beliefs support classical logic, and \( \mathcal{P} \) is a complete formal system for classical proof, then we are on very strong ground. The usual truth-table models for classical logic translate naturally into simple sorts of conceivable circumstances and absolutely concrete interpretations of primitive nonlogical symbols. But, if our customer’s beliefs support constructive logic, and \( \mathcal{P} \) is the Heyting calculus (which is complete for several well-known formal model-theoretic semantics), the translation of a formal countermodel into an intuitively convincing description is quite a bit harder.

In this paper, we compare the intuitive impact of three sorts of models for constructive logic: *Kripke/Beth* models based on possible worlds, L"uchli realizability models based on the formulae as types idea, and a new sort of model that we call *relational realizability* models, refining L"uchli’s ideas. There appear to be a number of different intuitive ideologies that lead to a belief in constructive logic. We do not insist on Brouwer’s, Heyting’s, Kolmogorov’s, Bishop’s or any other predetermined ideology supporting constructive logic. Rather, for each formal model theory, we seek the most natural intuitive explanation of it. We do not argue about which intuitive ideologies are metaphysically correct (perhaps each is correct for a different application of logical reasoning). Rather, we criticize the transparency with which each formal model theory is justified by its natural intuitive explanation. We find that this transparency improves from Kripke/Beth models through L"uchli models to relational realizability models, but it never approaches the clear connection between classical intuitions and truth-table models.

## 2 Formulae, Proofs, and Sequents

The rest of this paper discusses formulae, proofs of formulae, and models for formulae in the *positive first-order predicate calculus* and the *positive propositional calculus*. “Positive” means that we do not allow the logical negation symbol (\( \neg \)). Every one of the systems that we discuss can be extended easily to deal with negation, but in some cases that extension is somewhat subtle to understand—for example see [16] for a discussion of negation in L"uchli realizability models. We use a predicate calculus without function symbols and without equality. All