COMPLETENESS OF INTUITIONISTIC PROPOSITIONAL CALCULUS

by
Harvey Friedman
March 20, 2000

This is detailed proof of results we obtained in the 1980’s.

We give the uniform interpretation of HPC. We let HPC’, HQC’ be obtained by dropping negation and absurdity.

Define the type structure S(□) for every formula □ of HPC’, using Cartesian powers, products, and disjoint unions. S(A) for atoms A is taken to be N.

An assignment is a function f that assigns subsets of N to some atoms. Then f is extended to f* which sends every formula A of HPC to a subset of S(A).

We will prove that for any formula □ in HPC, □ is provable in HPC if and only if the intersection of f*(□) is nonempty.

We begin by constructing a particular Kripke model that is complete for formulas □ in HPC’.

We say that a set X of formulas in HPC’ without absurdity is saturated if and only if it is nonempty, logically closed, and any given disjunction lies in X if and only if at least one of the disjuncts lies in X.

Let K be the set of all saturated sets of formulas in HPC’.

The particular Kripke model M has frame (P(K),□). We define forcing for atoms as follows. Let p □ K.

1) any two elements of p are comparable under inclusion. Then p forces an atom if and only if the atom lies in the union of Y;
2) otherwise. Then p forces every atom.

LEMMA 1. For any nonempty p in M and □ in HPC’, p forces □ if and only if □ lies in an element of p or p has incomparable elements.

Proof: By induction on □.

1. □ is an atom. By construction.

2. □ □ □. Suppose p forces □ □ □. Then □ lies in an element of p or p has incomparable elements, and □ lies in an element of p or p has incomparable elements. If any two elements of p are comparable, then □,□ lie in an element of p. Hence □,□ lie in some common element of p, and so □ □ □ lies in some element of p.

Conversely suppose □ □ □ lies in an element of p or p has incomparable elements. If p has incomprable elements then p forces □ and p forces □, in which case p forces □ □ □. If □ □ □ lies in an element of p then □,□ lie in an element of p, in which case p forces □,□. Hence p forces □ □ □.

3. □ □. Suppose p forces □ □. Let p force □. Then □ lies in an element of p or p has incomparable elements. Hence □ □ lies in an element of p or p has incomparable elements.

Conversely, suppose □ □ lies in an element of p or p has incomparable elements. In the first case, □ lies in an element of p or □ lies in an element of p, in which case p forces □ or p forces □. In the second case, p forces □, and hence p forces □ □.

4. □ □ □. Suppose p forces □ □ □. We can assume that any two elements of p are comparable. Let X be the union of p. We can assume that □ □ □ □ X. Let X □ {□} □ Y be a saturated set of formulas in HPC′ which does not contain □. (The construction of Y requires that X is nonempty). Let q = p □ {Y}. Then any two elements of q are comparable. Also q forces □. Hence q forces □. Hence □ lies in an element of q. This is the desired contradiction.

Conversely, suppose □ □ □ lies in an element of p or p has incomparable elements. Let p □ q, q forces □. Then □ lies in an element of q or q has incomparable elements. In the former case, □ lies in an element of q, and so q forces □. In the latter case, q forces □. In either case q forces □. QED

LEMMA 2. Let □ be a formula in HPC′. ∅ forces □ if and only if □ is provable in HPC′. I.e., M is a complete Kripke model.

Proof: If ∅ forces □ then by Lemma 1, □ lies in every nonempty saturated set of formulas in HPC′. Hence □ is provable in HPC′. QED

LEMMA 3. There is a complete Kripke model whose frame is (P(K),⊇).

Proof: By Lemma 2, taking complements and using duality. QED

LEMMA 4. Let W □ P(K) be downward closed. There exists f:K □ PP(K) such that the following holds. For all S □ K, S □ W if and only if the f(x), x □ S, have a common element.

Proof: Take f(x) to be the set of all S □ W such that x □ S. Let S □ W. Then S lies in each f(x), x □ S. So the f(x), x □ S, all have a common element (namely x). On the other hand, suppose that the f(x), x □ S, all have the common element U. If S is empty then S □ W. So we may assume that S is nonempty. Note that for all x □ S, x □ U. I.e., S □ U. Also, since S is nonempty, U □ W. Hence S □ W. QED

LEMMA 5. There is a function H from K into assignments such that the following holds. For all p □ K and atoms A, p forces A if and only if the f*(A), f □ H[p], have a common element. The values of the assignments are subsets of P(K).

Proof: We can rewrite the conclusion as follows. For all p □ K and atoms A, p forces A if and only if the H(x)*(A), x □ p, have a common element.

For each atom A, let K_A = {p □ K: p forces A}. For each atom A, let f_A:E □ PP(E) be given by Lemma 2 for K_A. Define H on K by taking H(x) to be the assignment which maps each atom A to f_A(x); i.e., H(x)(A) = f_A(x).

Let p □ K and A be an atom. Then p forces A if and only if p □ K_A if and only if the f_A(x), x □ p, have a common element if and only if the H(x)(A), x □ p, have a common element. QED

A special Kripke model is a Kripke model (P(X),⊇,R), where X is the set of assignments, such that the following holds. For all p □ X and atoms A, p forces A if and only if the f(A), f □ p, have a common element.

LEMMA 6. There is a special Kripke model which is complete for HPC’.

Proof: The set of saturated sets has the same cardinality as the set of assignements. Identify X with E and apply Lemma 5. QED

LEMMA 7. Let (P(X),⊇,R) be a special Kripke model and □ be a formula of HPC′. For all p □ X, p forces □ in (P(X),⊇,R) if and only if the f*(□), f □ p, have a common element.

Proof: By induction on the formula □. The atomic case is true by the definition of special Kripke model. p forces □ □ □ if and only if p forces □ and p forces □ if and only if the intersection of f*(□) over f □ p is nonempty and the intersection of f*(□) over f □ p is nonempty if and only if f the intersection of f*(□ □ □) over f □ p is nonempty. p forces □   □ if and only if p forces □ or p forces □ if and only if the intersection of f*(□) over f □ p is nonempty or the intersection of f*(□) over f □ p is nonempty if and only if the intersection of f*(□   □) over f □ p is nonempty or the intersection of f*(□   □) over f □ p is nonempty. This latter argument crucially depends on the 0-1 signal in disjoint unions.

Suppose p forces □ □ □. Let x □ S(□). We now define H:S(□) □ S(□) as follows. Let q(x) = {f □ p: x □ f*(□)}. If q(x) is empty then set H(x) to be any element of S(□). Suppose q(x) is nonempty. Then the intersection of all f*(□), f □ q(x), is nonempty, since it includes x. Hence by induction hypothesis, q(x) forces □. Therefore q(x) forces □. By induction hypothesis, set H(x) to lie in the intersection of all f*(□), f □ q(x).

We now verify that H lies in all f*(□ □ □), f □ p. Let f □ p. We verify that H □ f*(□ □ □). To see this, let x □ S(□). Suppose x □ f*(□). We wish to show that H(x) □ f*(□).

Obviously q(x) is nonempty since it includes f. Hence H(x) lies in the intersection of all f*(□), f □ q(x). In particular, H(x) □ f*(□).

Finally suppose that the f*(□ □ □), f □ p, have a common element, H. Let q force □, p ⊇ q. By induction hypothesis, let x lie in the intersection of all f*(□), f □ q. We claim that H(x) lies in the intersection of all f*(□), f □ q. To see this, let f □ q. Then x □ f*(□). Hence H(x) □ f*(□).

By induction hypothesis, q forces □. Thus we have shown that p forces □ □ □. QED

This establishes the completeness theorem.

For full HPC, what we need to do is to use absurdity instead of negation, and treat it as just another atom, except that absurdity implies any atom. This is reflected in the semantics by simply requiring that the set assigned to absurdity is a subset of the set assigned to any atom.