Constructive Proof of Completeness for Proposition Logic

Robert Keller

11 October 2007

The Completeness Theorem says that for any set of formulas Γ and any formula φ,

$$\Gamma \models \varphi \text{ implies } \Gamma \vdash \varphi$$

i.e. if φ follows from Γ by truth-functional meaning (Γ *entails* φ), then φ can be derived from Γ by natural deduction reasoning. Although it is nice to have an arbitrary set of formulas Γ at our disposal, a given derivation can only use a finite set of formulas. So it is reasonable to consider the special case of this theorem where is Γ finite.

Once we agree to the finite Γ restriction, it is easy to reduce the theorem to the following special case, that Γcan be taken to be empty:

$$\models \varphi \text{ implies } \vdash \varphi$$

The reduction is by virtue of equivalence of the following type of statements:

a. $\{\varphi_1, \varphi_2, \ldots, \varphi_n\} \vdash \psi$
b. $\vdash \varphi_1 \rightarrow (\varphi_2 \rightarrow (\ldots \rightarrow (\varphi_n \rightarrow \psi) \ldots))$

In *a*, the formulas in a finite Γ are enumerated. In *b*, they are incorporated into a single formula. An analogous equivalence can be shown where $\vdash$ is replaced with $\models$.

Even with the simpler version, however, there doesn't seem to be a whole lot to go on to get a proof out of a tautology. We clearly need to use some kind of induction, and the structure of formulas seems to be the obvious place to try. But there is an issue of how to make it work. For example, consider the tautology

$$\neg p \vee p$$

Neither of the sub-parts of the formula is a tautology, so it is not clear how to use structural induction to show $\models \varphi$ implies $\vdash \varphi$.

As is often the case with inductive proofs, we need to prove something stronger that the desired result and observe that the desired result is a special case. The stronger assertion uses the concept of valuation, a function assigning a truth value to each proposition symbol in a formula. In general, a valuation will induce a truth value of 1 or 0 from the formula. As we know, a formula is a tautology iff every valuation induces the truth value 1.

We want to express a valuation ν in terms of a *collection of literals* (proposition symbols, or their negations). If ν is a valuation, define ν^* to be the set of literals having the value 1 under the valuation. That is, if $\nu(p) = 1$, then p appears in ν^*, while if $\nu(p) = 0$, then $\neg p$ appears in ν^*. For example, if $\nu(p) = 1$, $\nu(q) = 0$, $\nu(r) = 1$, then $\nu^* = \{p, \neg q, r\}$.

Lemma A: For any formula φ and valuation ν:

a. If ν(φ) = 1 then ν* ⊢ φ.
b. If ν(φ) = 0 then ν* ⊢ ¬φ.

In particular, if φ is a tautology, case *a* will always hold. We will then have a set of proofs of φ from all different assumptions ν*. We can then combine these proofs to get an absolute proof of ⊢ φ, giving the completeness theorem.

Proof of Lemma A: By structural induction on φ. Let ν be any valuation.

Basis: Suppose φ is a single proposition symbol.

If φ is ⊥, then ν(φ) = 0. So we need to prove ¬⊥. This is easly done: In a sub-proof having only one step, both assume and conclude ⊥. Then ¬⊥ follows in one step from the ¬I rule).

If φ is a proposition symbol p, then:
if ν(p) = 1, p is in ν*, so ν* ⊢ φ, in one step.
if ν(p) = 0, ¬p is in ν*, so ν* ⊢¬φ, also in one step.

Induction step: We consider the connectives ∧, →, and ¬. Considering ¬ separately from →⊥ is necessary to ensure the basis is eventually reached.

If φ is ψ ∧ ξ:

If ν(φ) = 1, then we must have ν(ψ) = ν(ξ) = 1. By the induction hypothesis, there are proofs of ν* ⊢ ψ and ν* ⊢ ξ. We can combine these proofs with one ∧Introduction to get a proof of φ.

If ν(φ) = 0, then we must have ν(ψ) = 0 or ν(ξ) = 0. Suppose it is the former, the latter being treatable symmetrically. By the induction hypothesis, there is a proof of ν* ⊢ ¬ψ. We can use a derived rule (1) for ¬ψ ⊢ ¬(ψ ∧ ξ) to get the proof of ¬φ.

If φ is ψ → ξ:

If ν(φ) = 0, then we must have ν(ψ) = 1 and ν(ξ) = 0. By the induction hypothesis, there are proofs of ν* ⊢ ψ and ν* ⊢ ¬ξ. We can use a derived rule (2) for ψ, ¬ξ ⊢ ¬(ψ → ξ) to complete the proof of φ.

If ν(φ) = 1, then we must have ν(ψ) = 0 or ν(ξ) = 1.

If $\nu(\xi) = 1$, then by the induction hypothesis, there is a proof of $\nu^* \vdash \xi$. Use $\rightarrow$Introduction in the degenerate form to get $\psi \rightarrow \xi$ to complete the proof of φ.

If $\nu(\psi) = 0$, then by the induction hypothesis, there is a proof of $\nu^* \vdash \neg\psi$. Use a derived rule (3) for $\neg\psi \vdash \psi \rightarrow \xi$ to complete the proof of φ.

If φ is $\neg\psi$:

If $\nu(\varphi) = 1$, then we must have $\nu(\psi) = 0$. By the induction hypothesis, there is a proof of $\nu^* \vdash \neg\psi$. Since φ *is* $\neg\psi$, this proof *is* a proof of φ.

If $\nu(\varphi) = 0$, then we need to consider separately the cases for ψ. In each case, $\nu(\psi) = 1$, so we prove $\nu^* \vdash \psi$, then use derived rule (4) $\psi \vdash \neg\neg\psi$, to get $\neg\varphi$.

If ψ is a proposition symbol p, $\nu(p) = 1$.

If ψ is $\xi \wedge \chi$, then $\nu(\xi) = \nu(\chi) = 1$. Use the induction hypothesis to get proofs of $\nu^* \vdash \xi$ and $\nu^* \vdash \chi$. Use $\wedge$Introduction to get a proof of ψ.

If ψ is $\xi \rightarrow \chi$, then $\nu(\chi) = 1$ or $\nu(\xi) = 0$.

In the first case, use the induction hypothesis to get a proof of $\nu^* \vdash \chi$. Then use the degenerate form of $\rightarrow$I to get $\xi \rightarrow \chi$.

In the second case, use the induction hypothesis to get a proof of $\nu^* \vdash \neg\xi$, and derived rule (2) to get $\neg\xi \vdash (\xi \rightarrow \chi)$.

This completes the proof of Lemma A.

Derived rule summary (proofs for the reader):

(1) $\neg\psi \vdash \neg(\psi \wedge \xi)$
(2) $\psi, \neg\xi \vdash \neg(\psi \rightarrow \xi)$
(3) $\neg\psi \vdash \psi \rightarrow \xi$
(4) $\xi \vdash \neg\neg\xi$

Proof of the Completeness Theorem: Suppose φ is a tautology. Let $p_1, p_2, \ldots, p_n$ be the proposition symbols occurring in φ. A proof of φ can be constructed by applying the following pattern *recursively* to the indicated sub-proofs

1	$p_1 \vee \neg p_1$	LEM (law of the excluded middle)
2	$[p_1]$. inner proof using the same pattern . φ	
3	$[\neg p_1]$. inner proof using the same pattern . φ	
4	φ	1, 2, 3, ∨E (or-elimination)

The *basis* of the construction is when all variables have been pulled out for sub-proofs, yielding a valuation that stipulates a value for every proposition symbol. In this case, we use the construction of Lemma A for the innermost sub-proofs.

Example: Consider the tautology φ = (p ∧ ¬q) → (¬q ∧ p). There are two proposition symbols p, q. The overall structure of the proof will be:

<table>
<tr><td>1</td><td>p ∨¬p</td><td>LEM</td></tr>
<tr><td>2</td><td colspan="2">
<table>
<tr><td>1</td><td>p</td><td>assumption</td></tr>
<tr><td>2</td><td>q ∨¬q</td><td>LEM</td></tr>
<tr><td>3</td><td colspan="2">
<table>
<tr><td>1</td><td>q</td><td>assumption</td></tr>
<tr><td></td><td>. . .</td><td></td></tr>
<tr><td></td><td>(p ∧ ¬q) → (¬q ∧ p)</td><td></td></tr>
</table>
</td></tr>
<tr><td>4</td><td colspan="2">
<table>
<tr><td>1</td><td>¬q</td><td>assumption</td></tr>
<tr><td></td><td>. . .</td><td></td></tr>
<tr><td></td><td>(p ∧ ¬q) → (¬q ∧ p)</td><td></td></tr>
</table>
</td></tr>
<tr><td>5</td><td>(p ∧ ¬q) → (¬q ∧ p)</td><td>2, 3, 4, ∨E</td></tr>
</table>
</td></tr>
<tr><td>3</td><td colspan="2">
<table>
<tr><td>1</td><td>¬p</td><td>assumption</td></tr>
<tr><td>2</td><td>q ∨¬q</td><td>LEM</td></tr>
<tr><td>3</td><td colspan="2">
<table>
<tr><td>1</td><td>q</td><td>assumption</td></tr>
<tr><td></td><td>. . .</td><td></td></tr>
<tr><td></td><td>(p ∧ ¬q) → (¬q ∧ p)</td><td></td></tr>
</table>
</td></tr>
<tr><td>4</td><td colspan="2">
<table>
<tr><td>1</td><td>¬q</td><td>assumption</td></tr>
<tr><td></td><td>. . .</td><td></td></tr>
<tr><td></td><td>(p ∧ ¬q) → (¬q ∧ p)</td><td></td></tr>
</table>
</td></tr>
<tr><td>5</td><td>(p ∧ ¬q) → (¬q ∧ p)</td><td>2, 3, 4, ∨E</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>(p ∧ ¬q) → (¬q ∧ p)</td><td>1, 2, 3, ∨E (or-elimination)</td></tr>
</table>

To complete the proof, we need to construct the inner-most sub-proofs according to Lemma A:

p, q ⊢ (p ∧ ¬q) → (¬q ∧ p)
p, ¬q ⊢ (p ∧ ¬q) → (¬q ∧ p)
¬p, q ⊢ (p ∧ ¬q) → (¬q ∧ p)
¬p, ¬q ⊢ (p ∧ ¬q) → (¬q ∧ p)

Proof of p, q ⊢ (p ∧ ¬q) → (¬q ∧ p):

The valuation has ν(p) = ν(q) = 1.
ν(¬q ∧ p) = 0, so we use ν(p ∧ ¬q) = 0.
We construct a proof of p, q ⊢ ¬(p ∧ ¬q), then use derived rule (3) for (p ∧ ¬q) → (¬q ∧ p).

To derive p, q ⊢ ¬(p ∧ ¬q), note that ν(¬(p ∧ ¬q)) = 1, so according to the handling of ¬∧, we use the proof of p, q ⊢ ¬¬q then derived rule (1) to get ¬(p ∧ ¬q).

Proof of $p, \neg q \vdash (p \wedge \neg q) \rightarrow (\neg q \wedge p)$:

The valuation has $\nu(p) = 1$, $\nu(q) = 0$.
$\nu(\neg q \wedge p) = 1$, so we derive $p, \neg q \vdash (\neg q \wedge p)$ using $\wedge$I, then $\rightarrow$I to get $(p \wedge \neg q) \rightarrow (\neg q \wedge p)$.

Proof of $\neg p, q \vdash (p \wedge \neg q) \rightarrow (\neg q \wedge p)$:

The valuation has $\nu(p) = 0$, $\nu(q) = 1$.
$\nu(\neg q \wedge p) = 0$, but $\nu(p \wedge \neg q) = 0$ and, so use derived rule (1) for $\neg p, q \vdash \neg(p \wedge \neg q)$, then derived rule (3) to get $(p \wedge \neg q) \rightarrow (\neg q \wedge p)$.

Proof of $\neg p, \neg q \vdash (p \wedge \neg q) \rightarrow (\neg q \wedge p)$:

The valuation has $\nu(p) = 0$, $\nu(q) = 0$.
$\nu(\neg q \wedge p) = 0$ and $\nu(p \wedge \neg q) = 0$, so use derived rule (1) for $\neg p, \neg q \vdash \neg(p \wedge \neg q)$, then derived rule (3) to get $(p \wedge \neg q) \rightarrow (\neg q \wedge p)$.

These are then put together in the overall proof:

1	$p \vee \neg p$	LEM
2	p	
3	$q \vee \neg q$	LEM
4	q	assumption
5	$\neg\neg q$	4, derived rule 4
6	$\neg(p \wedge \neg q)$	5, derived rule 1
7	$(p \wedge \neg q) \rightarrow (\neg q \wedge p)$	6, derived rule 3
8	$\neg q$	assumption
9	$(\neg q \wedge p)$	8, $\wedge$I
10	$(p \wedge \neg q) \rightarrow (\neg q \wedge p)$	9, $\rightarrow$I
11	$(p \wedge \neg q) \rightarrow (\neg q \wedge p)$	3, 4-7, 8-10, $\vee$E
12	$\neg p$	
13	$q \vee \neg q$	LEM
14	q	assumption
15	$\neg(p \wedge \neg q)$	14, derived rule 1
16	$(p \wedge \neg q) \rightarrow (\neg q \wedge p)$	15, derived rule 3
17	$\neg q$	assumption
18	$\neg(p \wedge \neg q)$	17, derived rule 1
19	$(p \wedge \neg q) \rightarrow (\neg q \wedge p)$	18, derived rule 3
20	$(p \wedge \neg q) \rightarrow (\neg q \wedge p)$	13, 14-16, 17-19, $\vee$E
21	$(p \wedge \neg q) \rightarrow (\neg q \wedge p)$	1, 2-11, 12-20, $\vee$E

Of course a much shorter proof is possible, but this was achieved by a method that is essentially algorithmic.

Below is the corresponding proof generated by my program, prover.pro:

```
Proof for tautology: implies(and(p, not(q)), and(not(q), p)):

| 1: or(p, not(p)) [lem]
| --------------------
| | 2: p [assumption(or-elim)]
| | 3: or(q, not(q)) [lem]
| | --------------------
| | | 4: q [assumption(or-elim)]
| | | 5: not(not(q)) [not-not-intro(4)]
| | | 6: not(and(p, not(q))) [derived1(5)]
| | | 7: implies(and(p, not(q)), and(not(q), p)) [derived3(6)]
| | --------------------
| | --------------------
| | | 8: not(q) [assumption(or-elim)]
| | | 9: and(not(q), p) [and-intro(8, 2)]
| | | 10: implies(and(p, not(q)), and(not(q), p)) [implies-intro(9)]
| | --------------------
| | 11: implies(and(p, not(q)), and(not(q), p)) [or-elim(3, 4-7, 8-10)]
| --------------------
| --------------------
| | 12: not(p) [assumption(or-elim)]
| | 13: or(q, not(q)) [lem]
| | --------------------
| | | 14: q [assumption(or-elim)]
| | | 15: not(and(p, not(q))) [derived1(12)]
| | | 16: implies(and(p, not(q)), and(not(q), p)) [derived3(15)]
| | --------------------
| | --------------------
| | | 17: not(q) [assumption(or-elim)]
| | | 18: not(and(p, not(q))) [derived1(12)]
| | | 19: implies(and(p, not(q)), and(not(q), p)) [derived3(18)]
| | --------------------
| | 20: implies(and(p, not(q)), and(not(q), p)) [or-elim(13, 14-16, 17-19)]
| --------------------
| 21: implies(and(p, not(q)), and(not(q), p)) [or-elim(1, 2-11, 12-20)]
```