Chapter 8: The Logic of Conditionals

§ 8.1 Informal methods of proof

Conditional elimination

This method of proof is also known by its Latin name, *modus ponens* (literally, "method of affirming"—roughly, having affirmed the antecedent of a conditional, you may affirm the consequent).

From P and P → Q , you may infer Q.

Biconditional elimination

This is sometimes called "*modus ponens* for the biconditional."

From P and P ↔ Q , you may infer Q.

From P and Q ↔ P , you may infer Q.

Some handy equivalences

Contraposition

P → Q ⇔ ¬Q → ¬P

The "conditional – disjunction" equivalence

P → Q ⇔ ¬P ∨ Q

The "negated conditional" equivalence

¬(P → Q) ⇔ P ∧ ¬Q

The "biconditional – conjunction" equivalence

P ↔ Q ⇔ (P → Q) ∧ (Q → P)

The "biconditional – disjunction" equivalence

P ↔ Q ⇔ (P ∧ Q) ∨ (¬P ∧ ¬Q)

In some systems of deduction for the propositional portion of FOL, these equivalences are used as rules. In the system $\mathcal{F}$, and in Fitch, these are **not** going to be rules. In fact, we will be using Fitch to prove these equivalences. Still, it is useful to be aware of them.

Note that the equivalence symbol ⇔ is not a connective of FOL, but a symbol we use in the "metalanguage" (in this case, English, the language in which we talk about FOL), and the statements of equivalence are not FOL sentences.

For a useful chart of tautological equivalences, see the [Supplementary Exercises](#) page on the course web site. Look under the listings for Chapter 7.

The method of conditional proof

This very important method of proof is a way of establishing **conditional** sentences. In using this method, we make a **provisional** assumption, *P*, and deduce some consequences of it. When we arrive at some appropriate sentence, *Q*, we have shown that the **assumption** of *P* has led to the conclusion *Q*. We may then **discharge** the assumption of *P* and conclude that *if P, then Q*.

How do we know when we have reached an "appropriate sentence, *Q*"? We should know in advance which sentence we are looking for. Since we are trying to establish a **conditional** sentence, we assume its antecedent and attempt to deduce its consequent.

Note that the method of conditional proof can be used for biconditionals, too. To prove $P \leftrightarrow Q$, construct separate conditional proofs for each of the conditionals $P \rightarrow Q$ and $Q \rightarrow P$. The conjunction of these two conditionals is equivalent to the biconditional $P \leftrightarrow Q$. (See the "biconditional – conjunction" equivalence above.)

§ 8.2 Formal rules of proof for $\rightarrow$ and $\leftrightarrow$

Conditional elimination ($\rightarrow$ Elim)

$$\begin{array}{l|l} & P \rightarrow Q \\ & P \\ \triangleright & Q \end{array}$$

That is, if you have a conditional on one line in a proof, and its antecedent (alone) on another line, you may infer the consequent. As justification, you cite the two earlier lines.

Conditional introduction ($\rightarrow$ Intro)

$$\begin{array}{l|l} & \begin{array}{|l} P \\ \hline Q \end{array} \\ \triangleright & P \rightarrow Q \end{array}$$

This is the formal counterpart of the method of conditional proof. Begin a subproof with P, the antecedent of your desired conditional. When your desired consequent, Q, occurs on a later line in the subproof, **end the subproof** and enter the conditional $P \rightarrow Q$. As justification for the conditional, you cite the **entire subproof**.

Some tricks using $\rightarrow$ Intro

Here are a couple of "tricks" involving conditionals, one involving an "irrelevant" antecedent and one an "irrelevant" consequent. Open the files Irrelevant Antecedent and Irrelevant Consequent on the Supplementary Exercises page. In each case you should be able to use a simple conditional proof strategy (with a little trick thrown in) to deduce the conclusion. (You probably used one of these strategies on Exercise 8.26.)

In these cases, what we have shown is that we can have any sentence we like as the antecedent of a conditional whose consequent we have already proved, and we can have any sentence we like as the consequent of a conditional if we've already proved the negation of its antecedent.

Here's a good problem on which to use the tricks you've just learned. Open Conditional Tricks on the Supplementary Exercises page. Then try to use these tricks in constructing a proof. (This is one half of the "negated conditional" equivalence we studied above; the proof you just constructed will make up half of the proof of that equivalence in Exercise 8.30.)

When you've finished the proof, **leave Fitch running** with your proof file still open. We'll be using it again in a moment.

Proofs without premises

It's easy to use → **Intro** to convert a proof with a premise into a proof (without premises) of the **corresponding conditional sentence**. The trick is just to **embed** the old proof as a subproof into the new proof.

Here's an easy way to embed on old proof into a new one. (This procedure is described in §4.4.3 of the software manual.) Open a new Fitch file, and start a new subproof (`Ctrl-P`). Now go back to the proof you've just finished, and click on the **rectangle** at the upper left of the window, to the left of the symbol selector—this converts the cursor into a selector than can copy many lines in a proof simultaneously.

Point the cursor at the upper left corner of the proof and click on the left mouse button. You will notice that the cursor has changed shape. Now, hold the button down while dragging toward the lower right corner. You will see a box appear, surrounding the text you've selected. When it encloses your entire proof, release the mouse button. You have just selected your entire proof.

Now click on `Edit Copy` (or type `Ctrl-C`) to copy the proof you've selected. Go back to your new proof; the focus slider should still be pointed at the assumption line of your new subproof. (If it's not, change the focus so that it is.) Then click on `Edit Paste` (or type `Ctrl-V`). This will insert your entire old proof into the new one, but one level of subproof deeper. Your old premise has become the first subproof assumption, and your old conclusion is the last line of that subproof. Don't forget to switch back to **pointer tool** (click on **arrow** at upper left).

Change the focus to the last line, and end the subproof. You are now prompted for a rule, so choose → **Intro** and cite the entire subproof. Then click on Check Step. Congratulations! You have just constructed a proof without premises. The old premise has become the first subproof assumption, and the old conclusion is the last line of that subproof. The last line (which has no premises) is the original argument's **corresponding conditional sentence**.

We will be looking at other (more complex) proofs without premises later.

Default and generous uses of the →rules

Default

If you are on a new line in a proof, and you choose rule → **Elim** and cite, as two separate lines, a conditional and its antecedent, Fitch will take the consequent and enter it by itself on the new line.
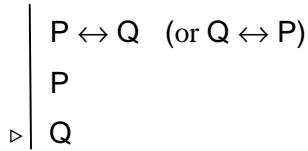
If you end a subproof, Fitch will create a new line after the subproof and ask you to choose a rule to justify the line. And as we saw (in our example of embedding a proof as a new subproof) when we chose → **Intro** and cited the entire subproof, Fitch entered, on the new line, the conditional sentence whose antecedent was the assumption of the subproof and whose consequent was the last line of the subproof.

Generous

In using → **Elim**, you can cite the support sentences in any order. In using → **Intro**, the consequent of the new conditional does not have to be the last line of the subproof—it can be any line of the subproof, even the assumption line itself.
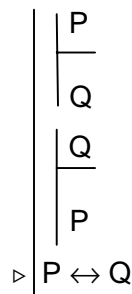
Biconditional elimination (↔ Elim)

P ↔ Q   (or Q ↔ P)

P

▷ Q

This rule is, effectively, *modus ponens* in either direction—given a biconditional on one line, and either of its components on another line, you may infer the other component on a new line.

Biconditional introduction (↔ Intro)

P

Q

Q

P

▷ P ↔ Q

This rule is, effectively, a double use of → **Intro**. If you have the two subproofs which would entitle you to infer, by → **Intro**, a pair of conditionals that are converses of one another (they have antecedent and consequent exchanged), you may infer the biconditional of the two component sentences.

§ 8.3 Soundness and completeness

In this course, we are mainly interested in developing a system of logic that we can use to prove the validity of valid arguments, and demonstrate the invalidity of invalid arguments. In more advanced logic courses, the attention turns to proving things **about** the system of logic itself—this is **metatheory**, the study of the properties of a logical system.

Two of the most important properties of such a system are **soundness** and **completeness**. We will not attempt to prove that our system (even the part of it we've developed so far) has these properties. Rather, we're just interested in understanding what these properties are, and with getting a rough idea of what would be involved in proving that our system has them. To study more metatheory, you'll have to go on to take PHIL 470.

Soundness and completeness are properties of the system $\mathcal{F}$—the set of deductive **rules** we've been developing. Actually, what we've studied so far is only a part of the system $\mathcal{F}$—the part that concerns the sentence connectives. That is, the system we've studied is just the collection of **Intro** and **Elim** rules for ¬, ∧, ∨, →, ↔, and ⊥. We'll call this smaller system $\mathcal{F}_T$. And we'll write:

$$P_1,…, P_n \vdash_T S$$

to mean that there is a proof in $\mathcal{F}_T$ of S from premises $P_1,…, P_n$.

Soundness

Basically, a **sound** system is one in which all the inferences that you are entitled to make by the rules of the system are valid inferences.

> The **Soundness Theorem** for $\mathcal{F}_T$: If $P_1,\ldots, P_n \vdash_T S$, then S is a tautological consequence of $P_1,\ldots, P_n$.

That is, the rules of $\mathcal{F}_T$ are sound if they satisfy this condition: any conclusion we can deduce from a set of premises by means of the rules of $\mathcal{F}_T$ is in fact a tautological consequence of those premises.

To put it another way, the soundness of $\mathcal{F}_T$ comes to this: if you can prove a conclusion using Fitch, it will also turn out to be a tautological consequence of its premises according to Boole.

A Corollary

Given the soundness of $\mathcal{F}_T$, we know that any conclusion we prove using its rules is indeed a tautological consequence of the premises we use to prove it. This also applies when there are **no premises**. That is, if there is a proof of S in $\mathcal{F}_T$ with no premises, then S is a tautology.

> **Soundness Corollary**: If $\vdash_T S$, then S is a tautology.

Completeness

Completeness is just the converse of soundness. That is, a **complete** system is one in which any valid inference can be proved by means of the rules of the system.

> The **Completeness Theorem** for $\mathcal{F}_T$: If a sentence S is a tautological consequence of $P_1,\ldots, P_n$, then $P_1,\ldots, P_n \vdash_T S$.

That is, the rules of $\mathcal{F}_T$ are complete if they satisfy this condition: any conclusion that is in fact a tautological consequence of a set of premises can be deduced from those premises by means of the rules of $\mathcal{F}_T$.

To put it another way, the completeness of $\mathcal{F}_T$ comes to this: if a conclusion is a tautological consequence of its premises according to Boole, then you can also prove it from those premises using Fitch.

A Corollary

Given the completeness of $\mathcal{F}_T$, we know that any tautological consequence of a given set of premises can be proved from those premises using the rules of $\mathcal{F}_T$. This also applies when there are **no premises**. That is, if S is a tautology, then there is a proof of S in $\mathcal{F}_T$ with no premises.

> **Completeness Corollary**: If S is a tautology, then $\vdash_T S$.

Why we need both soundness and completeness

It is not enough for a system to be sound, for a system can be sound without being complete. For example, a system whose only rules were ∧ **Intro** and ∧ **Elim** would be sound (it would never permit any invalid inferences), but woefully incomplete (it could not prove even the simplest inferences using connectives other than ∧.) So a system that is sound but not complete would never produce any incorrect inferences, but there would be some correct inferences it would be unable to prove.

Similarly, it is not enough for a system to be complete, for a system can be complete without being sound. For example, consider a "system" whose only rule is this: from P, infer Q. In this system, every inference would be simple: you could deduce any conclusion you want in just one step. Such a system would be too powerful; although it would include all the correct inferences, it would let in all the incorrect ones, too.

Proving soundness and completeness

Soundness is much easier to prove than completeness. To prove the soundness of the rules of $\mathcal{F}_T$, we only have to prove that none of the rules is capable of deducing a sentence that does not follow from (i.e., is not a TT-consequence of) its premises.

The method of proof is illustrated on pp. 215-216. It combines a proof by cases with a proof by contradiction. Assume that $\mathcal{F}_T$ is unsound. That means that there is a proof with an *invalid step*—a line in a proof that is licensed by the rules of $\mathcal{F}_T$ but that is not a tautological consequence of its premises. Pick any such proof, and consider the *first invalid step* in the proof. That step had to be licensed by one of the rules of $\mathcal{F}_T$. Then show that, no matter which rule we consider, the assumption that this rule has yielded an invalid step leads to a contradiction.

For example, here's how we show that → **Elim** could not have produced the first invalid step. Suppose that it did. Then we would have a pair of sentences Q and Q → R that are both tautological consequences of some set of premises $A_1 \ldots A_k$, but the sentence R obtained by → **Elim** is not a consequence of $A_1,\ldots, A_k$. But that means that a truth table for these sentences would contain a row in which $A_1,\ldots, A_k$, Q, and Q → R are all true, but R is false. But this is impossible—there can be no such row.

The full proof would go on to show that a similar impossibility results no matter which rule we suppose was the culprit that introduced the first invalid step.

The proof of completeness of $\mathcal{F}_T$ is more complicated, and will not be covered in this course. The proof can be found in §17.2, and by the time you have finished this course, you will be in a position to follow it.

§ 8.4 Valid arguments: some review exercises

We will work on Exercise 8.48.prf and Exercise 8.52.prf. for practice.

8.48

Since the conclusion is a conditional, we will want to use conditional proof strategy (→ **Intro**). So we will start a subproof with ¬Tet(c) as the assumption. But notice that the first premise is a conjunction, and we may need one of its conjuncts separately. (It's always a good idea to use ∧ **Elim** first, when possible, before setting up any subproofs. That way individual conjuncts will be available in all subsequent subroofs.) The second conjunct is a disjunction whose disjuncts occur in the other premises, so first we'll enter it on the next line (∧ **Elim**) and then set up our conditional proof. Open Proof 8.48a.prf.

The goal of our subproof is FrontOf(a, b), the consequent of the main goal. We can use proof by cases to get it. Open Proof 8.48b.prf. The first case is easy: Medium(b) gets us our conclusion by → **Elim**. The second case leads to a contradiction, since we can get both Tet(c) and ¬Tet(c).

That allows us to use ∨ **Elim** to complete the proof by cases of FrontOf(a, b). Then we use → **Intro** to complete our conditional proof of ¬Tet(c) → FrontOf(a, b). See Proof 8.48c.prf for the complete proof.

8.52

Open Exercise 8.52.prf. Again, the conclusion is a conditional, so we will set the proof up as a conditional proof (→ **Intro**). And we will use proof by contradiction (¬ **Intro**) in the embedded subproof to obtain a ≠ c, the consequent of the main conditional. Open Proof 8.52a.prf.

Dodec(b) is clearly inconsistent with Cube(b), so if we assume Cube(b) we can use **Ana Con** to derive ⊥ from that assumption. That lets us prove ¬Cube(b). Open Proof 8.52b.prf. Our task is now to get from ¬Cube(b) to a contradiction. But our premise is a biconditional, with Cube(b) as its left-hand component. Using **Taut Con** (remember, after p. 222 we're allowed to use **Taut Con** for "simple steps") we can infer the negation of the right-hand component. Then, substituting c for a (as licensed by a = c), we obtain a contradiction (the negation of a biconditional of the form P ↔ P) from which we can infer ⊥ by **Taut Con**. See the complete Proof 8.52c.prf.

Another example: a complicated tautology

We can use system $\mathcal{F}_T$ (and hence Fitch) to prove (without premises) any tautology. Here's an interesting example that looks as if it should be difficult to prove, but turns out to be very easy:

$$(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$$

The challenge is to prove this sentence from the empty set of premises. And the apparent difficulty is that the conclusion is long and complex, and there are no premises to work with.

The solution is to allow the logical structure of the conclusion to dictate the proof strategy. Since the conclusion is a conditional, we will plan to get it by using → **Intro**, assuming (A → (B → C)) and deriving (A → B) → (A → C). Here's what the opening strategy looks like. Open Ch8Ex1a.prf.

Now our task is to derive the intermediate conclusion (A → B) → (A → C). Since this is also a conditional, we can use the same → **Intro** strategy. We'll assume A → B and derive A → C. Open Ch8Ex1b.prf.

Our final intermediate conclusion is A → C, and we simply repeat the same strategy: assume the antecedent A and derive the consequent C. Open Ch8Ex1c.prf.

Now that the entire strategy has been laid out, we have several nested subproofs. Working inside the innermost subproof, our task is to derive C from all of the assumptions under which it falls. Since those assumptions are:

$$A \rightarrow (B \rightarrow C)$$
$$A \rightarrow B$$
$$A$$

the deduction is easy (three applications of → **Elim**). For the complete proof, see
Ch8Ex1d.prf.

Another example: a logical truth requiring Ana Con

In this example, we prove a logical truth that is neither a tautology nor a first-order logical truth—so we'll have to use **Ana Con**:

Cube(a) → ¬(Tet(a) ∨ Dodec(a))

Open the file Ch8Ex2.prf from the Supplementary Exercises page on the course web site. Notice that the Goal Constraints permit the use of **Ana Con**, but only with literals. Note carefully what this means. First, and most obviously, it means you can't apply **Ana Con** directly to the conclusion itself, justifying it in a single step. (Without this goal constraint, you could do this!). Second, and less obviously, you can't use **Ana Con** to justify ¬(Tet(a) ∨ Dodec(a)), citing the previous step Cube(a). For even though you would be applying **Ana Con** to a literal, the sentence you are deriving is not a literal. This constraint requires that **both** the cited sentence **and** the derived sentence must be literals.

Try to develop an approach that works within this constraint: my version uses conditional proof strategy, with an embedded indirect proof; the contradiction in the indirect proof is obtained by using proof by cases. Your version might well be different. For my version, see Proof Ch8Ex2.prf.