CMU 15-671 | Models of Software Systems | Fall 1997

Other Proof Techniques and Proof Hints

Garlan & Wing | Handout 3 | 10 September 1997

Natural deduction and equational reasoning, which you have seen already, and induction, which you will see shortly, are examples of different proof techniques to have in your bag of proof tricks. This handout covers some others. All but the first are probably familiar to you, which is why we are not going to dwell on any of them. Much of this handout is inspired by [GS]; parts are copied verbatim from Chapters 3 and 4.

1 One Point Rule

The One Point Rule is useful for dealing with existential quantifiers. It says that if you have an existentially quantified statement and part of it pins down an exact value for the quantified variable, then you can remove the quantification and *replace* the variable by the value whereever that variable appears in the rest of the statement.

$$\frac{\exists\, x : S \bullet (p \wedge x = t)}{t \in S \wedge p[t/x]} \qquad \neg(occurs(x, t))$$

Note the *side condition* that you must check before applying this rule: The variable, x, must not be free in the term, t, that you are using in the substitution.

For example, $\exists\, n : \mathbb{N} \bullet (n < 10 \wedge n = 2 * 3)$ is equivalent to $2 * 3 \in \mathbb{N} \wedge 2 * 3 < 10$. The first conjunct checks that the term is member of the right set; it is usually easy to discharge that check. Then, you are left with just the part of the predicate in which the substitution has been made, e.g., $2 * 3 < 10$.

2 Implication [GS]

If your predicate has an implication in it, sometimes it's easier to work with the equivalent disjunctive formulation:

$$p \Rightarrow q \equiv \neg p \vee q$$

Here are some other useful properties about implication and boolean constants:

$p \Rightarrow p \equiv true$ (**Reflexivity**)
$p \Rightarrow true \equiv true$ (**Right zero**)
$true \Rightarrow p \equiv p$ (**Left identity**)
$p \Rightarrow false \equiv \neg p$
$false \Rightarrow p \equiv true$

Here are some useful theorems about implication:

$p \wedge q \Rightarrow r \equiv p \Rightarrow (q \Rightarrow r)$ (**Shunting**)
$p \wedge (p \Rightarrow q) \equiv p \wedge q$
$p \vee (p \Rightarrow q) \equiv true$

3 Assuming the Antecedent [GS]

It is common in mathematics to prove an implication $p \Rightarrow q$ by assuming the antecedent p and proving the consequent q. By "assuming the antecedent" we momentarily think of it as an axiom and thus equivalent to *true*. The formal justification of this proof technique relies on the **Deduction Theorem** from mathematical logic.[1]

Here is a simple example (from [GS]): Suppose we want to prove $a \wedge b \Rightarrow (a = b)$. In the proof we get to assume the conjuncts of the antecedent. We are required to prove $a = b$:

Proof:

$$
\begin{array}{ll}
\textbf{Assume} & a, b. \\
& a \\
= & \quad \text{\% Assumption of a.} \\
& true \\
= & \quad \text{\% Assumption of b.} \\
& b
\end{array}
$$

Warning: For the proof of $p \Rightarrow q$, the **Deduction Theorem** requires that all variables in the assumed expression, p, be viewed as constants throughout the proof of q, so that you cannot substitute expressions in for the "variables" that appear in the assumptions. The following incorrect proof of $(b = c) \Rightarrow (d = c)$ (which is not valid) shows why this is necessary. The proof is incorrect because b in the assumption is replaced.

Proof (Incorrect):

$$
\begin{array}{ll}
\textbf{Assume} & b = c. \\
& d \\
= & \quad \text{\% Assumption of } (b = c)[d/b]\text{, i.e., } d = c \\
& c
\end{array}
$$

Notice that the expression in which we are attempting to do the substitution is $b = c$.

4 Case Analysis [GS]

A proof of r by case analysis proceeds as follows. Find the cases p and q such that $p \vee q$ holds. Then show that r holds in each case: $p \Rightarrow r$ and $q \Rightarrow r$. The justification of this proof technique relies on this property of implication:

$$(p \Rightarrow r) \wedge (q \Rightarrow r) \equiv (p \vee q \Rightarrow r) \qquad \textbf{(Case Analysis)}$$

Notice the common (and simpler) version of this property when $q \equiv \neg p$:

$$(p \Rightarrow r) \wedge (\neg p \Rightarrow r) \equiv r \qquad \textbf{(Simple Case Analysis)}$$

This proof technique generalizes to a case analysis of more than two cases in the obvious way: The disjunction of all the cases must be *true* and each case must imply r.

5 Proof by Contradiction [GS]

It is common in mathematics to prove a theorem p by assuming it is *false* and deriving a contradiction, i.e., derive *false* or something equivalent to *false*, like $q \wedge \neg q$. The justification for this proof technique is the following property about implication, repeated from Section 2:

[1] **Deduction Theorem**: Suppose adding $p_1, \ldots, p_n$ as axioms to propositional logic, with the variables of the p_i considered to be constants, allows q to be proved. Then $p_1 \wedge \ldots \wedge p_n \Rightarrow q$ is a theorem.

$p \Rightarrow \mathit{false} \equiv \neg p$

By substituting $\neg p$ for p and using double negation, we derive the theorem:

$\neg p \Rightarrow \mathit{false} \equiv p$ (**Proof by Contradiction**)

Hence, having proved that $\neg p \Rightarrow \mathit{false}$ is a theorem, you may conclude that p is a theorem.

6 Proof by Contrapositive [GS]

A typical proof of the implication $p \Rightarrow q$ might proceed as follows (notice a combination of proof techniques, assuming the antecedent and proof by contradiction):

- Assume p.
- Prove q by contradiction. That is, assume $\neg q$ and prove *false*.

This technique is really a proof by contrapositive in disguise:

$p \Rightarrow q \equiv \neg q \Rightarrow \neg p$ (**Contrapositive**)

7 Proof Hints

- Use the structure of an expression to guide what proof step to take next.
  - Look for subexpressions that you can replace with equivalent expresssions.
  - Use algebraic and logical properties of logical connectives and quantifiers to rewrite formulae. For example, use the One Point Rule to eliminate the existential; use properties of implication (Section 2) to get rid of implications; use properties about quantifiers to move them outward across connectives.
- Use lemmas and auxiliary definitions to add structure to your proof.
  - Introduce them to avoid repeating the same subexpressions or subproofs. (Sound familiar from your programming experience?)
  - Use them to make your proofs more readable, shorter, and more modular.
- Suppose you have an expression, e, given in terms of an operator f. (In fact you may introduced f yourself following the hint above.)
  - Expand the expression e by replacing f with its definition.
  - Then use properties about operators, g, in f's definition to manipulate and simplify the expanded e.
  - At the end, reintroduce (if convenient or necessary) the original f using its definition (in the other direction).