Additional Proof Techniques and Applications
CS 2LC3

Ryszard Janicki

Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada

Additional Proof Techniques

- When dealing with proofs of boolean expressions, our equational logic suffices.
- When dealing with other domains of interest (e.g. integers, sequences, or trees), where we use inductively defined objects, partial functions and the like, a few additional proof techniques become useful.
- In this section, we introduce these techniques.
- In doing so, we can begin looking at the relation between formal and informal proofs.

Assuming the Antecedent

- A common practice in mathematics is to prove an implication $P \implies Q$ by assuming the antecedent P and proving the consequent Q .
- By "assuming the antecedent" we mean thinking of it, momentarily, as an axiom and thus equivalent to true . In the proof of consequent Q , each variable in the new axiom P is treated as a constant

Theorem ((Extended) 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 $P_1 \wedge \ldots \wedge P_n \implies Q$ is a theorem.

Proof By Case Analysis

- A proof of P (say) by *case analysis* proceeds as follows.
- Find cases (boolean expressions) Q and R (say) such that $Q \vee R$ holds.
- Then show that P holds in each case: $Q \implies P$ and $R \implies P$.
- One could have a 3-case analysis, or a 4-case analysis, and so on; the disjunction of all the cases must be true and each case must imply P .

Prove: S
By cases: P, Q, R
(proof of $P \vee Q \vee R$ —omitted if obvious)
Case P**:** (proof of $P \Rightarrow S$)
Case Q**:** (proof of $Q \Rightarrow S$)
Case R**:** (proof of $R \Rightarrow S$)

Proof By Mutual Implication

- A proof by *mutual implication* of an equivalence $P \equiv Q$ is performed as follows:
- To prove $P \equiv Q$, prove $P \implies Q$ and $Q \implies P$.
- Such a proof rests on theorem (3.80), which we repeat here:

$$(p \implies q) \wedge (q \implies p) \equiv (p \equiv q).$$

Proof By Contradiction

- The formal basis: Theorem (3.74), $p \implies \mathit{false} \equiv \neg p$.
- Hence by substitution $p := \neg p$, we have **proof by contradiction**, i.e.

$$\neg p \implies \mathit{false} \equiv p.$$

- Proofs by contradiction cannot be used as basis for constructing algorithms. They usually state existence of some entity or property.

Proof by Contrapositive

- An implication $P \implies Q$ is sometimes proved as follows.
- First assume P ; then prove Q by contradiction, i.e. assume $\neg Q$ and prove false.
- Such a proof is not as clear as we might hope, and there is a better way:
- **Proof method:** Prove $P \implies Q$ by proving its *contrapositive* $\neg Q \implies \neg P$. (see (3.61)).

Applications

- *Statement in English*: If Joe fails to submit a project in course CS414, then he fails the course. If Joe fails CS414, then he cannot graduate. Hence, if Joe graduates, he must have submitted a project.
- *Formalisation*:
  s : Joe submits a project in CS414.
  f : Joe fails CS414.
  g : Joe graduates.
  $F0 : \neg s \implies f$, $F1 : f \implies \neg g$, $C : g \implies s$.
  We want $F0 \wedge F1 \implies C$., i.e.
  $(\neg s \implies f) \wedge (f \implies \neg g) \implies (g \implies s)$.
- *Proof*:

$$
\begin{array}{ll}
 & (\neg s \Rightarrow f) \wedge (f \Rightarrow \neg g) \\
\Rightarrow & \quad \langle \text{Transitivity of } \Rightarrow \ (3.82a) \rangle \\
 & \neg s \Rightarrow \neg g \\
= & \quad \langle \text{Contrapositive } (3.61) \rangle \\
 & g \Rightarrow s
\end{array}
$$

- Value v is in $b[1..10]$ means that if v is in $b[11..20]$ then it is not in $b[11..20]$.
- *Formalization*
  x: v is in $b[1..10]$
  y: v is in $b[11..20]$
  Hence: $x \equiv y \implies \neg y$. WE simplify it:

$$
\begin{array}{ll}
 & x \equiv y \Rightarrow \neg y \\
= & \quad \langle \text{Rewrite implication (3.59)} \rangle \\
 & x \equiv \neg y \vee \neg y \\
= & \quad \langle \text{Idempotency of } \vee \text{ (3.26)} \rangle \\
 & x \equiv \neg y
\end{array}
$$

- Back to English: " v is in $b[1..10]$ means that it is not in $b[11..20]$".

- Consider the following, which is a simplification of a situation in Shakespeare's *Merchant of Venice*.
- Portia has a gold casket and a silver casket and has placed a picture of herself in one of them.
- On the caskets, she has written the following inscriptions:
  Gold: The portrait is not in here.
  Silver: Exactly one of these inscriptions is true.
- Portia explains to her suitor that each inscription may be true or false , but that she has placed her portrait in one of the caskets in a manner that is consistent with this truth or falsity of the inscriptions.
- If he can choose the casket with her portrait, she will marry him.
- The problem for the suitor is to use the inscriptions (although they could be true or false) to determine which casket contains her portrait.

- *Formalization*. Variables:

  gc : The portrait is in the gold casket.
  sc : The portrait is in the silver casket.
  g : The portrait is not in the gold casket.
  (This the inscription on the gold casket.)
  s : Exactly one of g and s is *true*.
  (This the inscription on the silver casket.)

- *Facts*:
  $F0 : gc \equiv \neg sc$
  $F1 : g \equiv \neg gc$
  $F2 : s \equiv (s \equiv \neg g)$

- *Solution*:

$$
\begin{array}{ll}
 & s \equiv s \equiv \neg g \\
= & \langle \text{Symmetry of } \equiv \text{ (3.2) —so } \neg g \equiv s \equiv s \equiv \neg g \ \rangle \\
 & \neg g \\
= & \langle\ F1\,;\ \text{Double negation (3.12)} \rangle \\
 & gc
\end{array}
$$