Programming Languages: Imperative Program Construction 2. Propositional and Predicate Logic

Shin-Cheng Mu

Autumn Term, 2021

Materials of this part of the course are adapted from Gries and Schneider [GS93]. Axioms and theorems are numbered according to the book.

1 Synatx and Evaluation of Boolean Expressions

Syntax

- Boolean expressions are constructed from
  - constants $True$ and $False$,
  - boolean variables, which can be associated (only) with values $True$ and $False$,
  - boolean operators $\equiv$, $\not\equiv$, $\neg$, $\vee$, $\wedge$, $\Rightarrow$, and $\Leftarrow$.
- $True$ and $False$ are often called *boolean values*.
- A boolean expression is said to be of type boolean.

Unary Operators

		id	$\neg$	
$True$	$True$	$True$	$False$	$False$
$False$	$True$	$False$	$True$	$False$

Binary Operators

See Figure 1.

- $\wedge$: "and", also called *conjunction* (合取).
- $\vee$: "or", also called *disjunction* (析取).
- $=$: equality (等於). In this book we give it another symbol $\equiv$, called *equivalence* (等價), whose operands are called *equivalents*, for reasons to be explained later.
- $\neq$ and $\not\equiv$: inequality and inequivalence. Many people do not know that $\neq$ is also *xor* (exclusive or).
- $\Rightarrow$: *implication* (蘊含). Expression $b \Rightarrow c$ is read as "b implies c" or "if b then c", where b is the *antecedent* (前件、前事) and c the *consequent* (後件、後果).
  - Note that $b \Rightarrow c$ is $True$ when b is $False$.
- $\Leftarrow$: *consequence*. Expression $b \Leftarrow c$ is read a "b follows from c", where b is the *consequent* and c is the *antecedent*. $b \Leftarrow c$ is equivalent to $c \Rightarrow b$
  - Many people are not aware that $\Leftarrow$ occurs very often in proofs.
- $nand$ and nor stands for "not and" and "not or". Expression $b\ nand\ c$ is $\neg(b \wedge c)$ and $b\ nor\ c$ is $\neg(b \vee c)$. Useful when you study digital circuits.
- **precedence**: See the table in the handouts. Basically, $\wedge$ binds tighter than $\vee$, like $\times$ binds tighter than $+$. Note also that $=$ has a high precedence, while $\equiv$ has a ver low precedence.

2 Satisfiability, Validity, and Duality

Using Truth Tables to Evaluate Boolean Expressions

How do you evaluate $p \vee (q \wedge \neg r)$?

							$\equiv$		nand	$\not\equiv$					nor	
		$\vee$	$\Leftarrow$		$\Rightarrow$		$=$	$\wedge$		$\neq$						
T	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
T	F	T	T	T	T	F	F	F	T	T	T	T	F	F	F	F
F	T	T	T	F	F	T	T	F	T	T	F	F	T	T	F	F
F	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F

Figure 1: Binary Boolean Operators

p	q	r	$\neg r$	$q \wedge \neg r$	$p \vee (q \wedge \neg r)$
T	T	T	F	F	T
T	T	F	T	T	T
T	F	T	F	F	T
T	F	F	T	F	T
F	T	T	F	F	F
F	T	F	T	T	T
F	F	T	F	F	F
F	F	F	T	F	F

How do you determine whether $p \vee (q \wedge \neg r) = (p \vee q) \wedge (p \vee \neg r)$?

Satisfiability and Validity

- A boolean expression P is *satisfiable in state* S if its value is $True$ in state S;
- P is *satisfiable* (可滿足) if there exists some S in which P is satisfiable;
- and P is *valid* (有效) if it is satisfiable in every state.
- A valid expression is called a *tautology* (重言式、恆真式、套套邏輯).
- Tautologies are of interest to us, because they represent universal truth: some property holds, regardless of the context.

Satisfiability and Validity

- Example: $p \vee q$ is satisfible in any state having $(p, True)$ (or $(q, True)$), thus it is satisfible. It is not valid, however, since it is falsified in state $[(p, False), (q, False)]$.
- Examples of tautologies (valid expressions):

$True$	$True \equiv True$
$p \vee True$	$p \vee q \equiv q \vee p$
$p \vee \neg p$	$(p \equiv q) \equiv (q \equiv p)$
$\neg(p \wedge False)$	$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Semantics v.s. Syntax

- As mentioned before, we are mainly interested in tautologies, since they represent universal truth.
- How to determine whether an expression (e.g. $\neg(p \wedge q) \equiv \neg p \vee \neg q$) is a tautology?
  - One may build a truth table — in effect, try all possibilities, and see whether the expression evaluates to $True$ in all states.
- This is a *semantical approach*.

Semantics (語意)

- What an expression "means". Or, what the "value" of an expression is.
- In our current logic (*propositional logic*), each expression evaluates to either $True$ or $False$.
  - To be more precise, each expression is a function from states to $True$ or $False$.
  - Validity of an expression can be checked by enumerating all possible states and see whether the expression always evaluate to $True$.
- A good property of propositional logic: we can always decide whether an expression is valid.
- However, this approach does not extend to more complex logic, say, a logic that involves natural numbers.

A Syntactical (語法) Approach

- A *formal* system: a collection of symbols, and some rules to manipulate the symbols.
- A *calculus*: a formal system designed for reasoning; a method or process of reasoning by calculation with symbols.

Formal Deduction Systems for Logic

- There are a number of formal deduction systems for various kinds of logic.
  - A collection of *axioms* (公理) — things that are believed to be true without doubts.
  - A collection of rules that, given true propositions, guarantee to produce true propositions.
- To *prove* a theorem (show that it is valid) is to show that the theorem can be derived, from the axioms, using the given rules.
- Example of such systems: natural deduction, sequent calculus, etc.
  - You'd see natural deduction in my Functional Programming course since it is closely related to types.
- All of them are *syntactical* rules to manipulate expression with.
- After designing a formal system, one has to verify, against the *semantics*, that the system is sound, and preferably, complete.

Calculational Logic

- For a number of reasons, in this course we advocate a more algebraic formal system: *calculational logic*.
- The axioms are expressions like: $True \equiv (p \equiv p)$, $(p \equiv q) \equiv (q \equiv p)$, etc. There are a lot of them, and we will discuss them one bye one.
- The inference rules are mainly **Substitution** (1.1), **Transitivity** (1.4), and **Leibniz** (1.5).
- A *theorem* (定理) is either an axiom, or an expression that, using the inference rules, proved to be equal to an axiom or a previously proved theorem.

Reminder: Substitution, Transitivity, and Leibniz

(1.1) **Substitution**: $\dfrac{E}{E[v\backslash F]}$

(1.4) **Transitivity**: $\dfrac{X = Y \qquad Y = Z}{X = Z}$

(1.5) **Leibniz**: $\dfrac{X = Y}{E[z\backslash X] = E[z\backslash Y]}$

3 Equivalence and True

Conjunctional Equality

- What does it mean when I write $a = b = c$?
- We usually think of it as an abbreviation of $a = b \wedge b = c$ (which, by transitivity, also gives us $a = c$).
- It is a useful abbreviation to have. In $E_0 = E_1 = \ldots = E_n < E_{n+1}$ one immediately see that any two expressions in $E_0 \ldots E_n$ are equal, and they are all smaller than E_{n+1}. That is lots of information in one expression.

Associative Equality

- But, $(p = q) = r$, when the variables are boolean values, may also be interpreted as "evaluate $p = q$ to a boolean value, and compare the result with r".
- For example, $(False = False) = True$ evaluates to $True = True$, which is $True$.
- Surprisingly, this definition of equality is associative: $(p = q) = r$ is always equal to $p = (q = r)$.
- Associativity of equality will turn out to be very useful later: it also allows us to compactly represent lots of information in a short axiom.
- We thus denote it using a different symbol: $\equiv$.

Equality v.s. Equivalence

- When we write $=$ we mean ordinary, conjunctive equality (等於).
- Another symbol, $\equiv$, referred to as "equivalence" (等價), denotes equality in the associative sense.
  - $p \equiv q$ is read as "p equivals q".
- $((p \equiv q) \equiv r) \;=\; (p \equiv (q \equiv r))$.
  - We may thus just write $p \equiv q \equiv r$.
- It is convenience to assign it a very low precedence.
- Compare:
  - $False \equiv False \equiv True$ evaluates to $True$, while
  - $False = False = True$ is an abbreviation of $False = False \wedge False = True$, which evaluates to $False$.

3.1 Axioms Regarding Equivalence

Associativity (結合律) and Symmetry (對稱性) of $\equiv$

(3.1) **Axiom, Associativity of $\equiv$:**
$((p \equiv q) \equiv r) \ \equiv\ (p \equiv (q \equiv r))$

- As mentioned before, (3.1) allows us to write $p \equiv q \equiv r$.

(3.2) **Axiom, Symmetry of $\equiv$:** $p \equiv q \equiv q \equiv p$

- When read as $(p \equiv q) \equiv (q \equiv p)$, it allows us to freely swap positions of terms connected by $\equiv$.
- When read as $p \equiv (q \equiv q \equiv p)$, it allows us to rewrite $q \equiv q \equiv p$ to p, and vice versa.

Example of Associativity of $\equiv$

- Consider $m + n$ is Even $\equiv m$ is Even $\equiv n$ is even.
  - When parenthesized as $m + n$ is Even $\equiv$ (m is Even $\equiv n$ is even), it says that $m + n$ is even exacly when both m and n are even.
  - When parenthesized as ($m + n$ is Even $\equiv$ m is Even) $\equiv n$ is even, it says that adding n to m does not change the parity of m exacly when n is even.
- It actually covers four cases:
  - (($m+n$ is even) and (m is even) and (n is even)), or
  - (($m+n$ is odd) and (m is odd) and (n is even)), or
  - (($m+n$ is odd) and (m is even) and (n is odd)), or
  - (($m+n$ is even) and (m is odd) and (n is odd)).

  We can thus see how associativity makes one expression so concise and expressive!

Our First Proof

Task: prove $p \equiv p \equiv q \equiv q$.

Proof:

$p \equiv p \equiv q \equiv q$
$=$ { Symmetry of $\equiv$ (3.2) — replace $p \equiv q \equiv q$ by p }
$p \equiv p$
$=$ { Symmetry of $\equiv$ (3.2) — replace p by $q \equiv q \equiv p$ }
$p \equiv q \equiv q \equiv p$ $\square$

- The expression has been shown to be equal to an axiom (3.2). Thus it is proved.
- This is not the only possible proof. We use this example to demonstrate the use of associativity of $\equiv$.

Identity of $\equiv$

(3.3) **Axiom, Identity of $\equiv$:** $\mathit{True} \equiv p \equiv p$

- When read as $\mathit{True} \equiv (p \equiv p)$ it says that p always equivals itself, that is, $\equiv$ is reflexive.
- When read as $(\mathit{True} \equiv p) \equiv p$, it shows that True is an identity element of $\equiv$. [1] It allows us to remove occurrences of $\mathit{True} \equiv$ in an expression.
- Given (3.1) and (3.2), we also have $p \equiv (\mathit{True} \equiv p)$, $(p \equiv \mathit{True}) \equiv p$, etc.

Sequences of Equivalences

- Consider $p \equiv p \equiv q \equiv p \equiv r \equiv q$.
  - With (3.1) and (3.2) it can be transformed to $p \equiv p \equiv p \equiv q \equiv q \equiv r$.
  - With (3.1) and (3.3) we may simplify it to: $\mathit{True} \equiv p \equiv \mathit{True} \equiv r$.
  - which further simplifies to $p \equiv r$.
  - In general, in $P_0 \equiv P_1 \equiv \ldots P_n$, any P_i that occurs an even number of times is removed, while any P_j that occurs an odd number of times is replaced by a single occurrence.
- $P_0 \equiv P_1 \equiv \ldots P_n$ is True exactly when an even number of P_i are False.
  - By identity of $\equiv$ (3.3), each $\mathit{False} \equiv \mathit{False}$ can be rewritten to True.

[1] I is a *identity element* (單位元素), or a *unit* of a binary operation $\circ$ if $x \circ I = I \circ x = x$, for all x. The identity element of $+$ is 0, and that of $\cdot$ is 1.

Two Theorems

(3.4) $True$
(3.5) **Reflexivity of $\equiv$:** $p \equiv p$

- We prove (3.4) below:

$$
\begin{array}{ll}
 & True \\
= & \{ \text{ Identity of } \equiv \text{ (3.3), with } p := True \ \} \\
 & True \equiv True \\
= & \quad \{ \text{ Identity of } \equiv \text{ (3.3) } - \\
 & \qquad \text{replace the 2nd } True \ \} \\
 & True \equiv p \equiv p - \text{ Identity of } \equiv \text{ (3.3)} \quad \square
\end{array}
$$

- Prove (3.5) as an exercise!

3.2 Some Words on Proof Format

Proving a Theorem

- As mentioned before, to prove a theorem is to show that it equals an axiom or a previously established theorem.

$$
\begin{array}{ll}
 & P \\
= & \{ \text{ Property (i.j), with } p := X \ \} \\
 & P_1 \\
= & \{ \text{ Property (k.l), with } q := Y \ \} \\
 & \qquad : \\
= & P_n - \text{ Property (m.n)} \quad \square
\end{array}
$$

- "Property (i.j)" is the name and number of each theorem/axiom used. You can give only the name, or the number, when it is obvious.
- "with $p := X$" specifies how property (i,j) is instantiated. It indicates an application of **Substitution** (1.1). Can be omitted when obvious.
- If a sub-expression is rewritten by the hint, there is an implicit application of **Leibniz** (1.5)
- Chained equality is an application of **Transitivity** (1.4).
- P_n is an (instance of) established theorem (m,n). It can be omitted when it is obvious (e.g. when P_n is $True$).
- You can also proceed the entire calculation the other way round —- start from a theorem and reach P, depending on which direction is easier.

Proving an Equivalence

- When proving an equivalence $P \equiv Q$, we can use a simplified format:

$$
\begin{array}{ll}
 & P \\
= & \{ \text{ Property (i.j), with } p := X \ \} \\
 & P_1 \\
= & \{ \text{ Property (k.l), with } q := Y \ \} \\
 & \qquad : \\
= & Q \qquad \square
\end{array}
$$

- It can always be transformed to a proof in the original format:

$$
\begin{array}{ll}
 & P \equiv P \\
= & \{ \text{ Property (i.j), with } p := X \ \} \\
 & P \equiv P_1 \\
= & \{ \text{ Property (k.l), with } q := Y \ \} \\
 & \qquad : \\
= & P \equiv Q \qquad \square
\end{array}
$$

4 Negation, Inequivalence, and False

Definitions

Two axioms regarding negation. The first defines $\neg$ and $False$, and the second defines inequality $\not\equiv$.[2]

(3.15) **Axiom, Definition of** $False$:
$\neg p \equiv p \equiv False$

(3.10) **Axiom, Definition of** $\not\equiv$:
$(p \not\equiv q) \equiv \neg(p \equiv q)$

Theorems Relating $\equiv$, $\not\equiv$, $\neg$

See Figure 2.

Distributivity of $\neg$ over $\equiv$

[2] The textbook takes (3.8) and (3.9) as axioms. I choose (3.15) as an axiom, in my opinion a more useful property, following [?]. It allows both (3.8) and (3.9) to be theorems.

(3.8) $False \equiv \neg True$

(3.9) **Distributivity of $\neg$ over $\equiv$:** $\neg(p \equiv q) \equiv \neg p \equiv q$

(3.11) $\neg p \equiv q \equiv p \equiv \neg q$

(3.12) **Double negation :** $\neg\neg p \equiv p$

(3.13) **Negation of $False$:** $\neg False \equiv True$

(3.14) $(p \not\equiv q) \equiv \neg p \equiv q$

(3.16) **Symmetry of $\not\equiv$:** $(p \not\equiv q) \equiv (q \not\equiv p)$

(3.17) **Associativity of $\not\equiv$:** $((p \not\equiv q) \not\equiv r) \equiv (p \not\equiv (q \not\equiv r))$

(3.18) **Mutual associativity :** $((p \not\equiv q) \equiv r) \equiv (p \not\equiv (q \equiv r))$

(3.19) **Mutual interchangeability :** $p \not\equiv q \equiv r \ \equiv \ p \equiv q \not\equiv r$

Figure 2: Theorems Relating $\equiv$ and $\neg$.

Proving (3.9).

$$
\begin{array}{ll}
 & \neg(p \equiv q) \\
= & \quad \{ \text{ (3.15) } \} \\
 & p \equiv q \equiv False \\
= & \quad \{ \text{ Symmetry and Associativity of } \equiv \ \} \\
 & p \equiv False \equiv q \\
= & \quad \{ \text{ (3.15) } \} \\
 & \neg p \equiv q \qquad \Box
\end{array}
$$

Sequence of $\equiv$ and $\not\equiv$

- (3.11) is an important law that allows us to shunt $\neg$ (and thus $\not\equiv$) around.
- Mutual associativity (3.17) and (3.18) allows us to omit parentheses in sequence of $\equiv$ and $\not\equiv$.
- Mutual associativity (3.19) allows us to exchange adjacent $\equiv$ and $\not\equiv$.

Sequences of $\equiv$ and $\not\equiv$

- Consider $p \equiv p \equiv q \equiv \neg p \equiv r \equiv \neg q$.
  - With the associativity laws it can be transformed to $p \equiv p \equiv \neg p \equiv q \equiv \neg q \equiv r$.
  - With (3.15) we may simplify it to: $p \equiv False \equiv False \equiv r$.
  - which further simplifies to $p \equiv True \equiv r$, and $p \equiv r$.
- Moreover,
  - None or both of p and q is $True$: $p \equiv q$.
  - Exactly one of p and q is $True$: $p \not\equiv q$.
  - 0, 2, or 4 of p, q, r, s are $True$: $p \equiv q \equiv r \equiv s$.
  - 1 or 3 of p, q, r, s are $True$: $\neg(p \equiv q \equiv r \equiv s)$.

4.1 Heuristics in Proofs

Heuristic of Structural Matching

Identify applicable theorems by matching the structure of expressions or subexpressions. The operators that appear in a boolean expression and the shape of its subexpressions can focus the choice of theorems to be used in manipulating it.

Avoid Repeating the Same Expression.

Principle: structure proofs to avoid repeating the same expression on many lines.

Heuristic of Definition Elimination

To prove a theorem concerning an operator $\circ$ that is defined in terms of another, say $\bullet$, expand the definition of $\circ$ to arrive at a formula that contains $\bullet$; exploit propertis of $\bullet$ to manipulate the formula, and then (possibly) reintroduce $\circ$ using its definition.

5 Disjunction

Definitions

The disjunction operator $\vee$ is defined by the following

five axioms. [3]

(3.24) **Axiom, Symmetry of** $\vee$: $p \vee q \equiv q \vee p$

(3.25) **Axiom, Associativity of** $\vee$:
$$(p \vee q) \vee r \equiv p \vee (q \vee r)$$

(3.26) **Axiom, Idempotency of** $\vee$: $p \vee p \equiv p$

(3.27) **Axiom, Distributivity of** $\vee$ **over** $\equiv$:
$$p \vee (q \equiv r) \equiv p \vee q \equiv p \vee r$$

(3.28) **Axiom, Excluded Middle** : $p \vee \neg p$

Basic Properties of $\vee$ [4]

(3.29) **Zero of** $\vee$**:** $p \vee \mathit{True} \equiv \mathit{True}$

(3.30) **Identity of** $\vee$: $p \vee \mathit{False} \equiv p$

(3.31) **Distributivity of** $\vee$ **over** $\vee$:
$$p \vee (q \vee r) \equiv (p \vee q) \vee (p \vee r)$$

(3.32) $p \vee q \equiv p \vee \neg q \equiv p$

Example: proving (3.29) Zero of $\vee$

(3.29) $p \vee \mathit{True} \equiv \mathit{True}$

Proof.

$$\begin{array}{ll} & p \vee \mathit{True} \\ = & \{\ (3.3)\ \} \\ & p \vee (p \equiv p) \\ = & \{\ (3.27)\ \} \\ & p \vee p \equiv p \vee p \\ = & \{\ (3.3)\ \} \\ & \mathit{True} \end{array}$$

□

Example: proving (3.29) Zero of $\vee$

The proof could have gone the other way round. But the first step would be "pulling a rabbit out of a hat!"

Proof.

$$\begin{array}{ll} & \mathit{True} \\ = & \{\ (3.3)\ \} \\ & p \vee p \equiv p \vee p \\ = & \{\ (3.27)\ \} \\ & p \vee (p \equiv p) \\ = & \{\ (3.3)\ \} \\ & p \vee \mathit{True} \end{array}$$

□

Proof Heuristics and Principles

Heuristic: To prove $P \equiv Q$, transform the expression with the most structure into the other.

Principle: Structure the proof to minimize the number of rabbits pulled out of a hat — make each step seem obvious, based on the structure of the expression and the goal of the manipulation.

6 Conjunction

The Golden Rule

(3.35) **Axiom, Golden rule** : $p \wedge q \equiv p \equiv q \equiv p \vee q$

- We can see it as a definition of $\wedge$: $p \wedge q \equiv (p \equiv q \equiv p \vee q)$.
- Or $(p \equiv q) \equiv (p \wedge q \equiv p \vee q)$, meaning that p and q are equal if their conjunction and disjunction are equal.
- Or we may use it to transform $p \wedge q \equiv p$ to $q \equiv p \vee q$, and vice versa.
- It is the only axiom we need regarding $\wedge$.

Basic Properties of $\wedge$

[3] A binary operator $\circ$ is *idempotent* (等幂) if $x \circ x = x$ for all x. Multiplication $\cdot$ and addition $+$ are not idempotent, but $\vee$ and $\wedge$ are.

[4] Z is a *zero* of a binary operation $\circ$ if $x \circ Z = Z \circ x = Z$, for all x. The term comes from the fact that 0 is the zero of $\cdot$.

(3.36) **Symmetry of** $\wedge$: $p \wedge q \equiv q \wedge p$
(3.37) **Associativity of** $\wedge$:
$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
(3.38) **Idempotency of** $\wedge$: $p \wedge p \equiv p$
(3.39) **Identity of** $\wedge$: $p \wedge \mathit{True} \equiv p$
(3.40) **Zero of** $\wedge$: $p \wedge \mathit{False} \equiv \mathit{False}$
(3.41) **Distributivity of** $\wedge$ **over** $\wedge$:
$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge (p \wedge r)$
(3.42) **Contradiction** : $p \wedge \neg p \equiv \mathit{False}$

Theorems relating $\wedge$ and $\vee$

(3.43) **Absorption** : $(a)\ p \wedge (p \vee q) \equiv p$
$(b)\ p \vee (p \wedge q) \equiv p$
(3.44) **Absorption** : $(a)\ p \wedge (\neg p \vee q) \equiv p \wedge q$
$(b)\ p \vee (\neg p \wedge q) \equiv p \vee q$
(3.45) **Distributivity of** $\vee$ **over** $\wedge$:
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
(3.46) **Distributivity of** $\wedge$ **over** $\vee$:
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
(3.47) **De Morgan** : $(a)\ \neg(p \wedge q) \equiv \neg p \vee \neg q$
$(b)\ \neg(p \vee q) \equiv \neg p \wedge \neg q$

Propositions as Sets

- To make sense of and to memorize the absorption laws, it helps to know that propositions are isomorphic to sets of states that satisfy the proposition.
- That is, a boolean expression E can be seen as the set of states that satisfy E.
- Disjunction ($\vee$) is set union ($\cup$); conjunction ($\wedge$) is intersection ($\cap$); negation ($\neg$) is set compliment.
- Indeed, $p \cap (p \cup q)$ equals p. The same with other absorption laws.
- True is the set of all states — all states satisfy True. False is the empty set — nothing satisfies False.

Theorems relating $\wedge$ and $\equiv$

(3.48) $p \wedge q \equiv p \wedge \neg q \equiv \neg p$
(3.49) $p \wedge (q \equiv r) \equiv p \wedge q \equiv p \wedge r \equiv p$
(3.50) $p \wedge (q \equiv p) \equiv p \wedge q$
(3.51) **Replacement** :
$(p \equiv q) \wedge (r \equiv p) \ \equiv \ (p \equiv q) \wedge (r \equiv q)$

Alternative Definitions of $\equiv$ and $\not\equiv$

(3.52) **Definition of** $\equiv$: $p \equiv q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
(3.53) **Exclusive or** : $p \not\equiv q \equiv (\neg p \wedge q) \vee (p \wedge \neg q)$

Proving (3.37) Associativity of $\wedge$

(3.37) $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Proof

$$
\begin{array}{ll}
 & (p \wedge q) \wedge r \\
= & \quad \{\ (3.35)\ \} \\
 & p \equiv q \equiv p \vee q \equiv r \equiv (p \equiv q \equiv p \vee q) \vee r \\
= & \quad \{\ (3.27)\ \} \\
 & p \equiv q \equiv r \equiv p \vee q \equiv p \vee r \equiv q \vee r \equiv p \vee q \vee r
\end{array}
$$

But this property is useful on its own! We will prove it as a lemma.

(3.55)
$(p \wedge q) \wedge r \ \equiv$
$p \equiv q \equiv r \equiv p \vee q \equiv p \vee r \equiv q \vee r \equiv p \vee q \vee r$

Proof: as in the previous slide.

Proving (3.37) Associativity of $\wedge$, again

(3.37) $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Proof

$$
\begin{array}{ll}
 & p \wedge (q \wedge r) \\
= & \quad \{\ (3.36)\ \} \\
 & (q \wedge r) \wedge p \\
= & \quad \{\ (3.55)\ \} \\
 & q \equiv r \equiv p \equiv q \vee r \equiv q \vee p \equiv r \vee p \equiv q \vee r \vee p \\
= & \quad \{\ (3.24) \text{ and } (3.2)\ \} \\
 & p \equiv q \equiv r \equiv p \vee q \equiv p \vee r \equiv q \vee r \equiv p \vee q \vee r \\
= & \quad \{\ (3.55)\ \} \\
 & (p \wedge q) \wedge r
\end{array}
$$

Using Lemmas

Principle: Lemmas can provide structure, bring to light interesting facts, and ultimately shorten a proof. [5]

Heuristic

Theorems stated in terms of $\equiv$ can be parsed in many ways. Exploit this ability.

7 Implication

Definition of Implication

(3.57) **Axiom, Definition of implication** :

$$p \Rightarrow q \;\equiv\; p \vee q \equiv q$$

(3.58) **Axiom, Consequences** : $p \Leftarrow q \equiv q \Rightarrow p$

Implication is Set Inclusion

- To understand the definition (3.57), again it helps to see propositions as sets.
- $p \Rightarrow q$ is understood as "if a state is in p, then it is in q". That is, p is a subset of q.
- Certainly, $p \cup q$ equals q exacly when p is a subset of q.
- But there are other definitions too!

Rewriting Implication

(3.59) **Definition of implication** :

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

(3.60) **Definition of implication** :

$$p \Rightarrow q \;\equiv\; p \equiv p \wedge q$$

(3.61) **Contrapositive** : $p \Rightarrow q \;\equiv\; \neg q \Rightarrow \neg p$

[5] A *lemma* is an auxiliary theorem used in a proof of some other theorem. The difference between "lemma" and "theorem" is in the eye of the beholder. The theorem is the thing we are interested in; the lemma is just a small theorem needed in its proof. However, some lemmas turn out later to be important on its own.

A Puzzle Using Contraposition

Six pencils stand in a pencil vase. Some pencils has a rubber eraser on one end, some do not. It is also assumed that a pencil is either just sharpened, or used.

The vase is opaque, so you could only see one end of each pencil. The six pencils you see now are respectively

1. without an eraser,
2. with an eraser,
3. used,
4. sharp,
5. sharp, and
6. with an eraser.

Some one claims that "all pencils with an eraser are used." To verify her claim, how many pencils do you have to pull out of the vase to check?

Miscellaneous Theorems About Implication

(3.62) $p \Rightarrow (q \equiv r) \equiv p \wedge q \equiv p \wedge r$

(3.63) **Distributivity of $\Rightarrow$ over $\equiv$** :

$$p \Rightarrow (q \equiv r) \equiv p \Rightarrow q \equiv p \Rightarrow r$$

(3.64) $p \Rightarrow (q \Rightarrow r) \;\equiv\; (p \Rightarrow q) \Rightarrow (p \Rightarrow r)$

(3.65) **Shunting** : $p \wedge q \Rightarrow r \;\equiv\; p \Rightarrow (q \Rightarrow r)$

(3.66) $p \wedge (p \Rightarrow q) \equiv p \wedge q$

(3.67) $p \wedge (q \Rightarrow p) \equiv p$

(3.68) $p \vee (p \Rightarrow q) \equiv \mathit{True}$

(3.69) $p \vee (q \Rightarrow p) \equiv q \Rightarrow p$

(3.70) $p \vee q \Rightarrow p \wedge q \;\equiv\; p \;\equiv\; q$

Implication and Boolean Constants

(3.71) **Reflexivity of $\Rightarrow$** : $p \Rightarrow p \equiv \mathit{True}$

(3.72) **Right zero of $\Rightarrow$** : $p \Rightarrow \mathit{True} \equiv \mathit{True}$

(3.73) **Left identity of $\Rightarrow$** : $\mathit{True} \Rightarrow p \equiv p$

(3.74) $p \Rightarrow \mathit{False} \equiv \neg p$

(3.75) $\mathit{False} \Rightarrow p \equiv \mathit{True}$

Weakening, Strengthening, and Modus Ponens[6]

(3.76) **Weakening, Strengthening** :

$(a)\ p \Rightarrow p \vee q$

$(b)\ p \wedge q \Rightarrow p$

$(c)\ p \wedge q \Rightarrow p \vee q$

$(d)\ p \vee (q \wedge r) \Rightarrow p \vee q$

$(e)\ p \wedge q \Rightarrow p \wedge (q \vee r)$

(3.77) **Modus ponens** : $p \wedge (p \Rightarrow q) \Rightarrow q$

Forms of Case Analysis

(3.78) $(p \Rightarrow r) \wedge (q \Rightarrow r) \equiv (p \vee q \Rightarrow r)$

(3.79) $(p \Rightarrow r) \wedge (\neg p \Rightarrow r) \equiv r$

Proving Implication by Deduction

There is another way to prove $p \Rightarrow q$, inspired by natural deduction: if we can prove q, assuming that p is an established property, that is,

$$
\begin{array}{ll}
 & q \\
= & \{\ \ldots\ \} \\
 & \ldots \\
= & \{\ p\ \} \\
 & \ldots \\
= & \{\ \ldots\ \} \\
 & True
\end{array}
$$

then we have proved $p \Rightarrow q$. In practice, many implications are proved this way.

Mutual Implication and Transitivity

(3.80) **Mutual implication** :

$(p \Rightarrow q) \wedge (q \Rightarrow p) \ \equiv\ p \equiv q$

(3.81) **Antisymmetry** :

$(p \Rightarrow q) \wedge (q \Rightarrow p) \Rightarrow (p \equiv q)$

(3.82) **Transitivity** :

$(a)\ (p \Rightarrow q) \wedge (q \Rightarrow r) \Rightarrow (p \Rightarrow r)$

$(b)\ (p \equiv q) \wedge (q \Rightarrow r) \Rightarrow (p \Rightarrow r)$

$(c)\ (p \Rightarrow q) \wedge (q \equiv r) \Rightarrow (p \Rightarrow r)$

[6] If $P \Rightarrow Q$, we say that P is stronger than Q and Q is weaker than P. The strongest formula is $False$, and the weakest formula is $True$.

Proving Implication by Transitivity

Transitivity allows us to prove an implication $p \Rightarrow q$ in yet another way.

$$
\begin{array}{ll}
 & q \\
= & \{\ \ldots\ \} \\
 & \ldots \\
\Leftarrow & \{\ \ldots\ \} \\
 & \ldots \\
= & \{\ \ldots\ \} \\
 & p
\end{array}
$$

7.1 Leibniz's Rule as an Axiom

(3.83) **Axiom, Leibniz** :

$(e = f) \Rightarrow (E[z\backslash e] = E[z\backslash F])$

Rules of Substitution

(3.84) **Substitution** :

$(a)\ (e = f) \wedge E^z_e \ \equiv\ (e = f) \wedge E^z_F$

$(b)\ (e = f) \Rightarrow E^z_e \ \equiv\ (e = f) \Rightarrow E^z_F$

$(c)\ q \wedge (e = f) \Rightarrow E^z_e \ \equiv\ q \wedge (e = f) \Rightarrow E^z_F$

Replacing Variables by Boolean Constants

(3.85) **Replace by** $True$:

$(a)\ p \Rightarrow E^z_p \ \equiv\ p \Rightarrow E^z_{True}$

$(b)\ q \wedge p \Rightarrow E^z_p \ \equiv\ q \wedge p \Rightarrow E^z_{True}$

(3.86) **Replace by** $False$:

$(a)\ E^z_p \Rightarrow p \ \equiv\ E^z_{False} \Rightarrow p$

$(b)\ E^z_p \Rightarrow p \vee q \ \equiv\ E^z_{False} \Rightarrow p \vee q$

(3.87) **Replace by** $True$: $p \wedge E^z_p \ \equiv\ p \wedge E^z_{True}$

(3.88) **Replace by** $False$: $p \vee E^z_p \ \equiv\ p \vee E^z_{False}$

(3.89) **Shannon** :

$E^z_p \ \equiv\ (p \wedge E^z_{True}) \vee (\neg p \wedge E^z_{False})$

References

[GS93] D. Gries and F. B. Schneider. *A Logical Approach to Discrete Math.* Springer, October 22, 1993.

A Table of Precedence

1. $[x \backslash E]$ (substitution, highest precedence)
2. (function application)
3. unary prefix operators: $+ - \neg \#$
4. $\times$ / **div mod**
5. $+ - \cup \cap$
6. $\uparrow \downarrow$
7. $= < > \in \subset \subseteq \supset \supseteq$
8. $\vee \wedge$
9. $\Rightarrow \Leftarrow$
10. $\equiv$ (lowest precedence)

All nonassociative binary infix operators associate to the left, $\triangleleft$, and $\Rightarrow$, except for $\Rightarrow$, which associate to the right.

Some operators may have a slash / through them to denote negation — e.g. $b \not\equiv c$ is an abbreviation for $\neg(b \equiv c)$.