Formal Logic

Formal Logic

- Formal Logic
  - The study of systems of deductive argument in which symbols are used to represent precisely defined categories of expressions
  - Formal logic deals with the manipulation of truth values (i.e., arguments).
    - Given one truth we wish to find another truth
- Statement
  - Simple sentence that is either true or false
  - Combination of statements form an argument
  - Represented by letters in formal logic

Connectives

- Truth Tables
  - Conjunction ($\wedge$)
  - Disjunction ($\vee$)
  - Implication ($\rightarrow$)
  - Equivalence ($\leftrightarrow$)
  - Negation ($'$)
- Order of Precedence (in our text):
  - ( )
  - $'$
  - $\wedge$, $\vee$
  - $\rightarrow$
  - $\leftrightarrow$

Well-formed formula (WFF): A formula constructed in accordance with the syntactic rules of logical connectives.

Propositional Logic: The logic of propositions (atomic statements).

A	B	$A \rightarrow B$	$B \rightarrow A$	$(A \rightarrow B) \wedge (B \rightarrow A)$
T	T			
T	F			
F	T			
F	F			

Truth Tables Applied

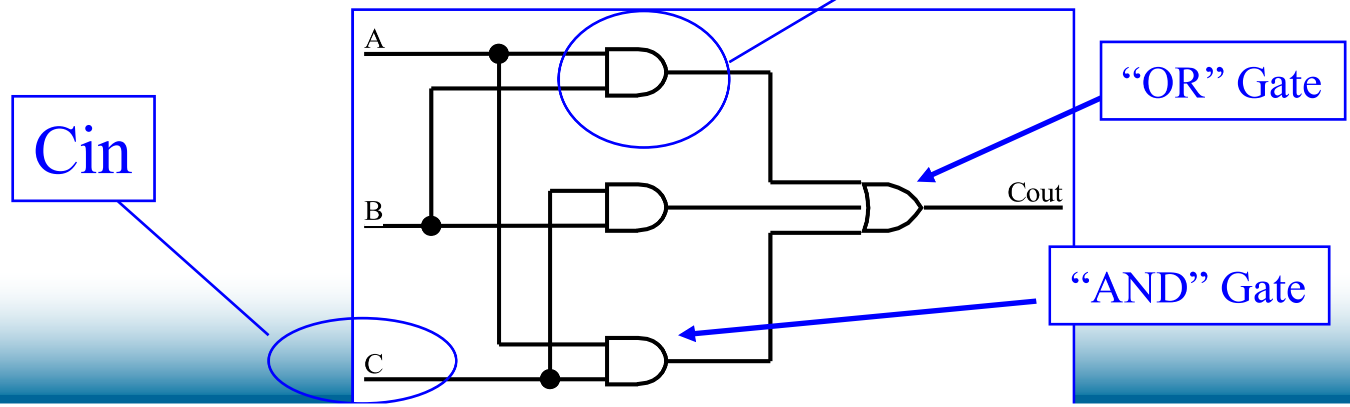
– Full Adder

Cout = ______________

Cin	A	B	S	**Cout**
F	F	F	F	**F**
F	F	T	T	**F**
F	T	F	T	**F**
F	T	T	F	**T**
T	F	F	T	**F**
T	F	T	F	**T**
T	T	F	F	**T**
T	T	T	T	**T**

A ∧ B handles both

– Gate Representation (Sum of Products):



Special WFFs

- Tautology
  - A WFF that is always true
  - "intrinsically true"
  - Example: $A \vee A'$
- Contradiction
  - A WFF that is always false
  - "intrinsically false"
  - Example: $A \wedge A'$

Both are important in matters of argument (i.e., proof).

The text shows several tautologies and equivalences of particular importance (p. 8)

Argument & Implication

- An argument has the form
  - Given x and y we can conclude z
  - What does this mean in logic?
  - When should this be considered a valid argument?
  - Hypothesis: When $x \wedge y \rightarrow z$ is true
    - Recall: $A \rightarrow B$ is true if A is false!
- Valid Argument
  - An argument $P \rightarrow Q$ is valid if it is a tautology.
  - Example: $((A \rightarrow B) \wedge A) \rightarrow B$

"modus ponens"

A	B	$A \rightarrow B$	$(A \rightarrow B) \wedge A$	$((A \rightarrow B) \wedge A) \rightarrow B$
T	T			
T	F			
F	T			
F	F			

Fallacy: an incorrect or misleading notion or opinion based on inaccurate facts or invalid reasoning

Rules of Inference

(a) Modus ponens (Latin: *mode that affirms*) is a valid, simple argument form

1. If p, then q
2. p
3. q

$$[(p \to q) \wedge p] \to q$$

(b) Modus tollens (Latin: *mode that denies*) is the formal name for indirect proof or proof by contrapositive

1. If p, then q
2. q′
3. p′

$$[(p \to q) \wedge q'] \to p'$$

Rules of Inference Exercise

1. If I go to Mars, I will run for office.
   I am going to Mars.
   Therefore, I will run for office.

2. If I go to Mars, I will run for office.
   I am not going to run for office.
   Therefore, I am not going to Mars.

3. If I go to Mars, I will run for office.
   I am going to run for office.
   Neither rule applies.

4. If I go to Mars, I will run for office.
   I am not going to Mars.
   Neither rules applies.

Rules of Inference

Two Deductively **Invalid** Rules of Inference

(a) Affirming the Consequent

1. If p, then q
2. q
3. p

(b) Denying the Antecedent

1. If p, then q
2. p′
3. q′

Rules of Inference (complete)

1. $P, P \to Q$	then	Q	(Modus Ponens, MP)
2. $Q', P \to Q$	then	P'	(Modus Tollens MT)
3. P, Q	then	$P \wedge Q$	(Conjunction)
4. $P \wedge Q$	then	P, Q	(Simplification)
5. P	then	$P \vee Q$	(Disjunction or Add)

They do not work in both directions, i.e., take rule 1:

Q then $P, P \to Q$ is false

Only works for equivalence rules (next slide)

Equivalence Rules

1. Commutative:

$P \vee Q$	is equiv. to	$Q \vee P$
$P \wedge Q$	$\equiv$	$Q \wedge P$

2. Associative:

$(P \vee Q) \vee R$	$\equiv$	$P \vee (Q \vee R)$
$(P \wedge Q) \wedge R$	$\equiv$	$P \wedge (Q \wedge R)$

3. De Morgan's:

$(P \vee Q)'$	$\equiv$	$P' \wedge Q'$
$(P \wedge Q)'$	$\equiv$	$P' \vee Q'$

Equivalence Rules

4. Implication:

$$P \to Q \quad \equiv \quad P' \vee Q$$

5. Double Negation:

$$P \quad \equiv \quad (P')'$$

6. Definition of equivalence:

$$P \leftrightarrow Q \quad \equiv \quad (P \to Q) \wedge (Q \to P)$$

To prove an argument:

- Do we need to generate the truth table to test if it is a tautology?
  - No.
- Use truth-maintaining derivations
  - Turn an existing wff into an equivalent wff
    - Called *equivalence rules*
    - Works in both directions
    - Page 24 of text
  - Add a new wff based on existence of other wffs
    - Called *inference rules*
    - Does not work in "reverse"
    - Pages 25 of text

"inference" – the act of concluding (a state of affairs, supposition, etc.) by reasoning from evidence

Characteristics of a Proof Sequence

- Given P_1 P_2 …….$P_n \rightarrow Q$
- Use formal logic to prove that Q is a valid conclusion from P_1,….P_n, with the following proof sequence:

1. P_1 (hypothesis)
2. P_2 (hypothesis)

:

3. P_n (hypothesis)
4. wff_1 (derived from earlier wffs – see next slide)

:

5. wff_m (derived from earlier wffs)
6. Q (derived from earlier wffs)

Argument & Implication

- Sample proof sequence
  - $A' \wedge (B' \vee A) \rightarrow B'$ is a valid argument

More Examples

Prove that the following argument is valid:

$$A \wedge (B \to C) \wedge [(A \wedge B) \to (D \vee C')] \wedge B \to D$$

1. Determine which term is the conclusion.
2. Separate each term to the left of $\to$ as a hypothesis.

$$A \wedge (B \to C) \wedge [(A \wedge B) \to (D \vee C')] \wedge B \to D$$

There are four hypotheses (separated by $\wedge$)

(a) A
(b) $(B \to C)$
(c) $((A \wedge B) \to (D \vee C')$
(d) B

Examples (cont'd)

Look at the obvious (stated) hypothesis and try to apply inference rules to get more truths:

(1) A	hyp
(2) $(B \to C)$	hyp
(3) $((A \wedge B) \to (D \vee C'))$	hyp
(4) B	hyp

New truths are:

(5) C	MP 2, 4
(6) $(A \wedge B)$	Conj 1, 4
(7) $(D \vee C')$	MP 3,6

What is the objective?

To derive the conclusion – in this case D.

Examples (cont'd)

We need to resolve D, so far, we have:

(1) A	hyp
(2) $(B \to C)$	hyp
(3) $((A \wedge B) \to (D \vee C'))$	hyp
(4) B	hyp
(5) C	MP 2, 4
(6) $(A \wedge B)$	Conj 1,4
(7) $(D \vee C')$	MP 3,6

From (7), note that $D \vee C'$ is true.

In other words, $C' \vee D$ is true, using the commutative rule.

(8) $C' \vee D$	7, commutative
(9) $C \to D$	8, imp
(10) D	9,5, mp

Law of Exportation (exp)

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

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

$$[(A \wedge B) \rightarrow C] \rightarrow [A \rightarrow (B \rightarrow C)]$$

Rules of Inference & Equivalence

From	Can Derive	Name
$P, P \to Q$	Q	MP
$P \to Q, Q'$	P'	MT
P, Q	$P \wedge Q$	CON
$P \wedge Q$	P, Q	SIM
P	$P \vee Q$	ADD

Expression	Equivalent to	Name
$P \vee Q$ $P \wedge Q$	$Q \vee P$ $Q \wedge P$	COM M
$(P \vee Q) \vee R$ $(P \wedge Q) \wedge R$	$P \vee (Q \vee R)$ $P \wedge (Q \wedge R)$	ASSO
$(P \vee Q)'$ $(P \wedge Q)'$	$P' \wedge Q'$ $P' \vee Q'$	DM
$P \to Q$	$P' \vee Q$	IMP
P	$(P')'$	DN
$P \leftrightarrow Q$	$(P \to Q) \wedge (Q \to P)$	EQU
$(P \wedge Q) \to R$	$P \to (Q \to R)$	EXP

Law of Hypothetical Syllogism (hs)

A.k.a. Transitive Law of Implication

Given $A \rightarrow B$ and $B \rightarrow C$

then $A \rightarrow C$

$$[(A \rightarrow B) \wedge (B \rightarrow C)] \rightarrow (A \rightarrow C)$$

Rules of Inference & Equivalence

From	Can Derive	Name
$P, P \to Q$	Q	MP
$P \to Q, Q'$	P'	MT
P, Q	$P \wedge Q$	CON
$P \wedge Q$	P, Q	SIM
P	$P \vee Q$	ADD

Expression	Equivalent to	Name
$P \vee Q$ $P \wedge Q$	$Q \vee P$ $Q \wedge P$	COM M
$(P \vee Q) \vee R$ $(P \wedge Q) \wedge R$	$P \vee (Q \vee R)$ $P \wedge (Q \wedge R)$	ASSO
$(P \vee Q)'$ $(P \wedge Q)'$	$P' \wedge Q'$ $P' \vee Q'$	DM
$P \to Q$	$P' \vee Q$	IMP
P	$(P')'$	DN
$P \leftrightarrow Q$	$(P \to Q) \wedge (Q \to P)$	EQU
$(P \wedge Q) \to R$	$P \to (Q \to R)$	EXP

More Inference Rules

- Disjunctive Syllogism: $[(P \vee Q) \wedge P'] \rightarrow Q$
- Contraposition: $[P \rightarrow Q] \rightarrow (Q' \rightarrow P')$
- Contraposition: $[Q' \rightarrow P'] \rightarrow [P \rightarrow Q]$
- Self-reference: $P \rightarrow (P \wedge P)$
- Self-reference: $(P \vee P) \rightarrow P$
- Inconsistence: $P, P' \rightarrow Q$
- Distributive: $P \wedge (Q \vee R) \leftrightarrow (P \wedge Q) \vee (P \wedge R)$
- Distributive: $P \vee (Q \wedge R) \leftrightarrow (P \vee Q) \wedge (P \vee R)$

Rules of Inference & Equivalence

From	Can Derive	Name
$P, P \to Q$	Q	MP
$P \to Q, Q'$	P'	MT
P, Q	$P \wedge Q$	CON
$P \wedge Q$	P, Q	SIM
P	$P \vee Q$	ADD
$P \to Q, Q \to R$	$P \to R$	HS
$P \vee Q, P'$	Q	DS
$P \to Q$	$Q' \to P'$	CONT
$Q' \to P'$	$P \to Q$	CONT
P	$P \wedge P$	SELF
$P \vee P$	P	SELF
P, P'	Q	INC

Expression	Equivalent to	Name
$P \vee Q$ $P \wedge Q$	$Q \vee P$ $Q \wedge P$	COMM
$(P \vee Q) \vee R$ $(P \wedge Q) \wedge R$	$P \vee (Q \vee R)$ $P \wedge (Q \wedge R)$	ASSO
$(P \vee Q)'$ $(P \wedge Q)'$	$P' \wedge Q'$ $P' \vee Q'$	DM
$P \to Q$	$P' \vee Q$	IMP
P	$(P')'$	DN
$P \leftrightarrow Q$	$(P \to Q) \wedge (Q \to P)$	EQU
$(P \wedge Q) \to R$	$P \to Q \to R$	EXP
$P \wedge (Q \vee R)$ $P \vee (Q \wedge R)$	$(P \wedge Q) \vee (P \wedge R)$ $(P \vee Q) \wedge (P \vee R)$	DIST

Practice Problems

- Mathematical Structures for Computer Science
  - Section 1.1: 7, 16, 17, 26, 32
  - Section 1.2: 10, 13, 17, 19, 20, 33, 35, 37, 38

Predicate Logic

- Propositional logic is quite limiting
  - Only speaks in terms of atomic objects
  - No properties to the objects
  - No relationships to other objects
- Clearly not realistic
- *Predicate logic* uses explicit predicates for representing properties of objects and relations between objects
- Predicate logic also adds the concept of variables and quantification

Predicate Logic

Example: All men are mortal.

Properties: man, mortal

Rewording: "Everything who has the property of being a man also has the property of being a mortal."

Predicate Logic

$$(\forall x)\ (\ \mathbf{man}(x) \rightarrow \mathbf{mortal}(x)\ )$$

Note: $\forall x$ and → almost always belong together.

Compare to:

$$(\forall x)\ (\ \mathbf{man}(x) \wedge \mathbf{mortal}(x)\ )$$

What does this translation mean?

Example: Some days are not rainy

Relations: day, rainy

Predicate Logic

$$(\exists x)\ (\ \mathbf{day}(x) \wedge \mathbf{rainy}(x)'\ )$$

Predicate Logic - Translations

– Example: Everybody loves somebody
  - Relations: loves, person
  - Translations:

$$\forall x\ \exists y\ (\ (person(x) \wedge person(y)) \rightarrow loves(x,y)\ )$$

Requires two people to exist before any “love” can happen.

$$\forall x\ (\ person(x) \rightarrow \exists y\ (person(y) \wedge loves(x,y)\ )\ )$$

Allows the inference of the existence of second person from the existence of a single person

$$\forall x\ \exists y\ loves(x,y)$$

Assumes only people are involved

Predicate Logic - Translations

P(x) = "x is a person"
T(x) = "x is a time"
F(x,y) = "x is fooled at y"

You can fool some of the people all of the time

"There exists at least one <u>person</u> such that for each unit of <u>time</u> that person can be <u>fooled</u>."

$$\exists x\ [P(x) \land \forall y\ [T(y) \rightarrow F(x,y)]]$$

You can fool all of the people some of the time

"For every <u>person</u> there exists at least one unit of <u>time</u> at which this person can be <u>fooled</u>."

$$\forall x\ [P(x) \rightarrow \exists\ y\ [\ T(y) \land F(x,y)]$$

Predicate Logic - Translations

You can't fool all of the people all of the time

"It is <u>not</u> the case that for every <u>person</u> and every unit of <u>time</u>, the person can be <u>fooled</u> at that time."

$$[\forall xy\ [(P(x) \wedge T(y)) \rightarrow F(x,y)]'$$

You can't fool all of the people all of the time

"There exists some person and some unit of time such that person is not fooled at that time."

$$\exists xy\ [(P(x) \wedge T(y) \wedge F(x,y)']$$

Predicate Logic

- Interpretation
  - Propositional was easy – each piece is either true or false
  - Predicates add complexity
  - An interpretation consists of:
    - A collection of objects, called the *domain* which must include at least one object
    - An assignment of a property of the objects in the domain to each predicate in the expression
    - An assignment of a particular object in the domain to each constant symbol in the expression
  - An universally quantified expression is true if the interpretation is true for all objects in the domain.
  - An existentially quantified expression is true if the interpretation is true for at least one set of objects in the domain.
  - Note: Nesting of quantifiers defines *scope*
  - *Free* variables are possible

Means we have picked some unspecified value x in the domain

Predicate Logic

- Valid Arguments in Predicate Logic
  - Potentially infinite domain means the truth table approach is impossible in general
  - Unlike with propositional logic, it has been proven that no algorithm exists for deciding validity of a predicate logic argument
  - How do we prove a predicate logic argument valid?
    - By using truth-maintaining transformations
  - The equivalence rules and inference rules for propositional logic still apply
    - Example:

$$\forall x\ R(x) \wedge [\forall y\ R(y) \rightarrow \forall z\ S(z)] \rightarrow \forall z\ S(z)$$

1.	$\forall x\ R(x)$	hyp
2.	$\forall y\ R(y) \rightarrow \forall z\ S(z)$	hyp
3.	$\forall z\ S(z)$	1,2,mp

A *validity* is a predicate wff that is true for all possible interpretations (analogous to tautology)

Predicate Logic

– Valid Arguments in Predicate Logic

- Four additional derivational rules
  - Universal Instantiation (ui)
  - Existential Instantiation (ei)
  - Universal Generalization (ug)
  - Existential Generalization (eg)

p. 50 of text

Apply to top-level formula

- Basic proof technique
  - Strip off quantifiers
  - Manipulate wffs
  - Put quantifiers back in

Of course, there are restrictions

- Example: $\exists x\ \exists y\ P(x,y) \rightarrow \exists y\ \exists x\ P(x,y)$

1.	$\exists x\ \exists y\ P(x,y)$	hyp
2.	$\exists y\ P(a,y)$	1,ei
3.	$P(a,b)$	2,ei
4.	$\exists x\ P(x,b)$	3,eg
5.	$\exists y\ \exists x\ P(x,y)$	4,eg

Predicate Logic

– Fallacy Example

$$\forall x \exists y P(x,y) \rightarrow \exists y \forall x P(x,y)$$

1.	$\forall x \exists y\ P(x,y)$	hyp
2.	$\exists y\ P(x,y)$	1,ui
3.	$P(x,b)$	2,ei
4.	$\forall x\ P(x,b)$	3,ug
5.	$\exists y \forall x\ P(x,y)$	4,eg

P(x,b) was deduced by ei from a wff in which x is free – NOT ALLOWED

Also not allowed to use ug if P(x) is deduced from any hypothesis in which x is free.

See page 50 of text!

Practice Problems

- Mathematical Structures for Computer Science
  - Section 1.3: 7, 8, 10, 11, 14, 29, 30

Application: Proof of Correctness

- How do we know if our programs are right?
  - Why do we care?
    - "This isn't medicine, no one is going to die"
  - What do you mean by "right"?
    - Verification – "Am I building the program right?"
    - Validation – "Am I building the right program?"
  - Proof of Correctness deals with Verification, not Validation

- When is a program right (or correct)?
  - If it produces the correct output for every possible input
    - Two parts:
      - » Program terminates
      - » Output is correct

Application: Proof of Correctness

- Definition: *partially correct*
  - A program, P, is partially correct with respect to an initial assertion q and the final assertion r if whenever q is true for the input values of P, then r is true for the output values of P.
- Notation: `{q}P{r}` called *Hoare Triple*
  - Note: This has nothing to do with whether a statement `P` halts
- In formal logic, this is equivalent to $\rightarrow$
  - $\forall x\ q(x) \rightarrow r(x, P(x))$
- To prove that a program is correct, we must assert the desired *postcondition* and prove that it is true within the assumption that the *precondition* is true and the *program* has been performed

Application: Proof of Correctness

– We must therefore know how each program structure transforms postconditions into preconditions
  - In other words, we need rules of inference that allow us to move from a given postcondition and program to an implied precondition

– Assignment Rule of Inference
  - if $\{R_i\}$ ***x* = *e***$\{R_{i+1}\}$
  - then R_i is R_{i+1} with *e* substituted for each *x*
  - Example:

```
{x == y + 2}
   x := x - 2;
{x == y}
```

Application: Proof of Correctness

- Proof of Correctness in Action
  - Do people really use this technique for large programs?
    - Yes and No
    - They use this technique on small, critical sections
    - They use a testing framework motivated, in part, by this technique (JUnit)